APPENDIX **A**

Subnetting Practice: 25 Subnetting Questions

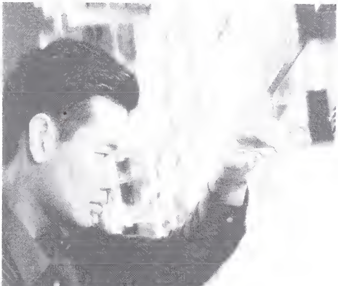
This appendix lists 25 separate questions, asking you to derive the subnet number, broadcast address, and range of valid IP addresses. In the solutions, the binary math is shown, as is the process that avoids binary math using the "subnet chart" described in Chapter 12, "IP Addressing and Subnetting." You might want to review Chapter 12's section on IP addressing before trying to answer these questions.

25 Subnetting Questions

Given each IP address and mask, supply the following information for each of these 25 examples:

- Size of the network part of the address
- Size of the subnet part of the address
- Size of the host part of the address
- The number of hosts per subnet
- The number of subnets in this network
- The subnet number
- The broadcast address
- The range of valid IP addresses in this network:

1. 10.180.10.18, mask 255.192.0.0
2. 10.200.10.18, mask 255.224.0.0
3. 10.100.18.18, mask 255.240.0.0
4. 10.100.18.18, mask 255.248.0.0
5. 10.150.200.200, mask 255.252.0.0
6. 10.150.200.200, mask 255.254.0.0
7. 10.220.100.18, mask 255.255.0.0
8. 10.220.100.18, mask 255.255.128.0



9. 172.31.100.100, mask 255.255.192.0
10. 172.31.100.100, mask 255.255.224.0
11. 172.31.200.10, mask 255.255.240.0
12. 172.31.200.10, mask 255.255.248.0
13. 172.31.50.50, mask 255.255.252.0
14. 172.31.50.50, mask 255.255.254.0
15. 172.31.140.14, mask 255.255.255.0
16. 172.31.140.14, mask 255.255.255.128
17. 192.168.15.150, mask 255.255.255.192
18. 192.168.15.150, mask 255.255.255.224
19. 192.168.100.100, mask 255.255.255.240
20. 192.168.100.100, mask 255.255.255.248
21. 192.168.15.230, mask 255.255.255.252
22. 10.1.1.1, mask 255.248.0.0
23. 172.16.1.200, mask 255.255.240.0
24. 172.16.0.200, mask 255.255.255.192
25. 10.1.1.1, mask 255.0.0.0

Suggestions on How to Attack the Problem

If you are ready to go ahead and start answering the questions, go ahead! If you want more explanation of how to attack such questions, refer back to the section on IP subnetting in Chapter 12. However, if you have already read Chapter 12, a reminder of the steps in the process to answer these questions, with a little binary math, is repeated here:

Step 1 Identify the structure of the IP address.

a. Identify the size of the network part of the address, based on Class A, B, and C rules.

b. Identify the size of the host part of the address, based on the number of binary 0s in the mask. If the mask is "tricky," use the chart of typical mask values to convert the mask to binary more quickly.

c. The size of the subnet part is what's "left over"; mathematically, it is 32 – (network + host)

d. Declare the number of subnets, which is $2^{\text{number-of-subnet-bits}} - 2$.

e. Declare the number of hosts per subnet, which is $2^{\text{number-of-host-bits}} - 2$

Step 2 Create the subnet chart that will be used in steps 3 and 4.

a. Create a generic subnet chart.

b. Write down the decimal IP address and subnet mask in the first two rows of the chart.

c. If an easy mask is used, draw a vertical line between the 255s and the 0s in the mask, from top to bottom of the chart. If a hard mask is used, draw a box around the interesting octet.

d. Copy the address octets to the left of the line or the box into the final four rows of the chart.

Step 3 Derive the subnet number and the first valid IP address.

a. On the line on the chart where you are writing down the subnet number, write down 0s in the octets to the right of the line or the box.

b. If the mask is difficult, so that there is a box in the chart, use the magic number trick to find the decimal value of the subnet's interesting octet, and write it down. Remember, the magic number is found by subtracting the interesting (non-0 or 255) mask value from 256. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

c. To derive the first valid IP address, copy the first three octets of the subnet number, and add 1 to the fourth octet of the subnet number.

Step 4 Derive the broadcast address and the last valid IP address for this subnet.

a. Write down 255s in the broadcast address octets to the right of the line or the box.

b. If the mask is difficult, so that there is a box in the chart, use the magic number trick to find the value of the broadcast address's interesting octet. In this case, you add the subnet number's interesting octet value to the magic number, and subtract 1.

c. To derive the last valid IP address, copy the first three octets of the broadcast address and subtract 1 from the fourth octet of the broadcast address.

Question 1: Answer

The answers begin with the analysis of the three parts of the address, the number of hosts per subnet, and the number of subnets of this network using the stated mask. The binary math for subnet and broadcast address calculation follows. The answer finishes with the easier mental calculations using the subnet chart described in Chapter 12.

Table A-1 *Question 1: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Item	**Example**	**Rules to Remember**
Address	10.180.10.18	N/A
Mask	255.192.0.0	N/A
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	22	Always defined as number of binary 0s in mask
Number of subnet bits	2	32 – (network size + host size)
Number of subnets	$2^2 - 2 = 2$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{22} - 2 = 4{,}194{,}302$	$2^{\text{number-of-host-bits}} - 2$

The binary calculations of the subnet number and broadcast address are in Table A-2. To calculate the two numbers, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table A-2 *Question 1: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.180.10.18	0000 1010 10**11 0100 0000 1010 0001 0010**
Mask	255.192.0.0	1111 1111 11**00 0000 0000 0000 0000 0000**
AND result (subnet number)	10.128.0.0	0000 1010 10**00 0000 0000 0000 0000 0000**
Change host to 1s (broadcast address)	10.191.255.255	0000 1010 10**11 1111 1111 1111 1111 1111**

To get the first valid IP address, just add 1 to the subnet number; to get the last valid IP address, just subtract 1 from the broadcast address. In this case:

10.128.0.1 through 10.191.255.254

10.128.0.0 + 1= 10.128.0.1

10.191.255.255 – 1= 10.191.255.254

Steps 2, 3, and 4 in the process use a table like Table A-3, which lists the way to get the same answers using the subnet chart and magic math described in Chapter 12. Figure A-1 at the end of this problem shows the fields in Table A-3 that are filled in at each step in the process. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table A-3 *Question 1: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4	Comments
Address	10	180	10	18	N/A
Mask	255	192	0	0	N/A
Subnet number	10	128	0	0	Magic number = 256 – 192 = 64
First address	10	128	0	1	Add 1 to last octet of subnet
Broadcast	10	191	255	255	128 + 64 – 1 = 191
Last address	10	191	255	254	Subtract 1 from last octet

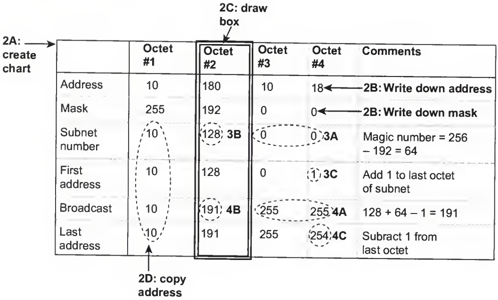
Subnet rule: Multiple of magic number closest to, but not more than, IP address value in interesting octet
Broadcast rule: Subnet + magic – 1

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 192 = 64 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 128 is the multiple of 64 that's closest to 180 but not bigger than 180. So, the second octet of the subnet number is 128.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 12, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 128 + 64 – 1 = 191.

Finally, Figure A-1 shows Table A-3 with comments about when each part of the table was filled in, based on the steps in the process at the beginning of the chapter.

Figure A-1 *Steps 2, 3, and 4 for Question 1*



	Octet #1	Octet #2	Octet #3	Octet #4	Comments
Address	10	180	10	18	**2B: Write down address**
Mask	255	192	0	0	**2B: Write down mask**
Subnet number	10	128 **3B**	0	0 **3A**	Magic number = 256 – 192 = 64
First address	10	128	0	1 **3C**	Add 1 to last octet of subnet
Broadcast	10	191 **4B**	255	255 **4A**	128 + 64 – 1 = 191
Last address	10	191	255	254 **4C**	Subract 1 from last octet

Question 2: Answer

Table A-4 *Question 2: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	10.200.10.18	N/A
Mask	255.224.0.0	N/A
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	21	Always defined as number of binary 0s in mask
Number of subnet bits	3	32 – (network size + host size)
Number of subnets	$2^3 - 2 = 6$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{21} - 2 =$ 2,097,150	$2^{\text{number-of-host-bits}} - 2$

Table A-5 presents the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in bold print in the table.

Table A-5 *Question 2: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.200.10.18	0000 1010 110**0 1000 0000 1010 0001 0010**
Mask	255.224.0.0	1111 1111 111**0 0000 0000 0000 0000 0000**
AND result (subnet number)	10.192.0.0	0000 1010 110**0 0000 0000 0000 0000 0000**
Change host to 1s (broadcast address)	10.223.255.255	0000 1010 110**1 1111 1111 1111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.192.0.1 through 10.223.255.254

Table A-6 lists the way to get the same answers using the subnet chart and magic math described in Chapter 12. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table A-6 *Question 2: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	**Octet 2**	**Octet 3**	**Octet 4**	**Comments**
Address	10	200	10	18	N/A
Mask	255	224	0	0	N/A
Subnet number	10	192	0	0	Magic number = 256 – 224 = 32
First address	10	192	0	1	Add 1 to last octet of subnet
Broadcast	10	223	255	255	192 + 32 – 1 = 223
Last address	10	223	255	254	Subtract 1 from last octet

Subnet rule: Multiple of magic number closest to, but not more than, IP address value in interesting octet
Broadcast rule: Subnet + magic – 1

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 224 = 32 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 192 is the multiple of 32 that's closest to 200 but not bigger than 200. So, the second octet of the subnet number is 192.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 12, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 192 + 32 – 1 = 223.

Question 3: Answer

Table A-7 *Question 3: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	10.100.18.18	N/A
Mask	255.240.0.0	N/A
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	20	Always defined as number of binary 0s in mask
Number of subnet bits	4	32 – (network size + host size)
Number of subnets	$2^4 - 2 = 14$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{20} - 2 = 1{,}048{,}574$	$2^{\text{number-of-host-bits}} - 2$

The binary calculations of the subnet number and broadcast address are in Table A-8. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table A-8 *Question 3: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.100.18.18	0000 1010 0110 **0100 0001 00100001 0010**
Mask	255.240.0.0	1111 1111 1111 **0000 0000 0000 0000 0000**
AND result (subnet number)	10.96.0.0	0000 1010 0110 **0000 0000 0000 0000 0000**
Change host to 1s (broadcast address)	10.111.255.255	0000 1010 0110 **1111 1111 1111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.96.0.1 through 10.111.255.254

Table A-9 lists the way to get the same answers using the subnet chart and magic math described in Chapter 12. Remember, subtracting the interesting (non-0 or 255) mask

value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table A-9 *Question 3: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4	Comments
Address	10	100	18	18	N/A
Mask	255	240	0	0	N/A
Subnet number	10	96	0	0	Magic number = 256 – 240 = 16
First address	10	96	0	1	Add 1 to last octet of subnet
Broadcast	10	111	255	255	96 + 16 – 1 = 111
Last address	10	111	255	254	Subtract 1 from last octet

Subnet rule: Multiple of magic number closest to, but not more than, IP address value in interesting octet
Broadcast rule: Subnet + magic – 1

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 240 = 16 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 96 is the multiple of 16 that's closest to 100 but not bigger than 100. So, the second octet of the subnet number is 96.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 12, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 96 + 16 – 1 = 111.

Question 4: Answer

Table A-10 *Question 4: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	**Example**	**Rules to Remember**
Address	10.100.18.18	N/A
Mask	255.248.0.0	N/A
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	19	Always defined as number of binary 0s in mask
Number of subnet bits	5	32 – (network size + host size)
Number of subnets	$2^5 - 2 = 30$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{19} - 2 = 524{,}286$	$2^{\text{number-of-host-bits}} - 2$

The binary calculations of the subnet number and broadcast address are in Table A-11. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table A-11 *Question 4: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.100.18.18	0000 1010 0110 0**100 0001 00100001 0010**
Mask	255.248.0.0	1111 1111 1111 1**000 0000 0000 0000 0000**
AND result (subnet number)	10.96.0.0	0000 1010 0110 0**000 0000 0000 0000 0000**
Change host to 1s (broadcast address)	10.103.255.255	0000 1010 0110 0**111 1111 1111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.96.0.1 through 10.103.255.254

Table A-12 lists the way to get the same answers using the subnet chart and magic math described in Chapter 12. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table A-12 *Question 4: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4	Comments
Address	10	100	18	18	N/A
Mask	255	248	0	0	N/A
Subnet number	10	96	0	0	Magic number = 256 – 248 = 8
First address	10	96	0	1	Add 1 to last octet of subnet
Broadcast	10	103	255	255	96 + 8 – 1 = 103
Last address	10	103	255	254	Subtract 1 from last octet

Subnet rule: Multiple of magic number closest to, but not more than, IP address value in interesting octet
Broadcast rule: Subnet + magic – 1

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 248 = 8 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 96 is the multiple of 8 that's closest to 100 but not bigger than 100. So, the second octet of the subnet number is 96.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 12, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 96 + 8 – 1 = 103.

Question 5: Answer

Table A-13 *Question 5: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	**Example**	**Rules to Remember**
Address	10.150.200.200	N/A
Mask	255.252.0.0	N/A
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	18	Always defined as number of binary 0s in mask
Number of subnet bits	6	32 – (network size + host size)
Number of subnets	$2^6 - 2 = 62$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{18} - 2 = 262{,}142$	$2^{\text{number-of-host-bits}} - 2$

The binary calculations of the subnet number and broadcast address are in Table A-14. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table A-14 *Question 5: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.150.200.200	0000 1010 1001 01**10 1100 1000 1100 1000**
Mask	255.252.0.0	1111 1111 1111 11**00 0000 0000 0000 0000**
AND result (subnet number)	10.148.0.0	0000 1010 0110 00**00 0000 0000 0000 0000**
Change host to 1s (broadcast address)	10.151.255.255	0000 1010 0110 01**11 1111 1111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.148.0.1 through 10.151.255.254

Table A-15 lists the way to get the same answers using the subnet chart and magic math described in Chapter 12. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table A-15 *Question 5: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4	Comments
Address	10	150	200	200	N/A
Mask	255	252	0	0	N/A
Subnet number	10	148	0	0	Magic number = 256 – 252 = 4
First address	10	148	0	1	Add 1 to last octet of subnet
Broadcast	10	151	255	255	148 + 4 – 1 = 151
Last address	10	151	255	254	Subtract 1 from last octet

Subnet rule: Multiple of magic number closest to, but not more than, IP address value in interesting octet
Broadcast rule: Subnet + magic – 1

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 252 = 4 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 148 is the multiple of 4 that's closest to 150 but not bigger than 150. So, the second octet of the subnet number is 148.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 12, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 148 + 4 – 1 = 151.

Question 6: Answer

Table A-16 *Question 6: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	**Example**	**Rules to Remember**
Address	10.150.200.200	N/A
Mask	255.254.0.0	N/A
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	17	Always defined as number of binary 0s in mask
Number of subnet bits	7	32 – (network size + host size)
Number of subnets	$2^7 - 2 = 126$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{17} - 2 = 131,070$	$2^{\text{number-of-host-bits}} - 2$

The binary calculations of the subnet number and broadcast address are in Table A-17. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in bold print in the table.

Table A-17 *Question 6: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.150.200.200	0000 1010 1001 011**0 1100 1000 1100 1000**
Mask	255.254.0.0	1111 1111 1111 111**0 0000 0000 0000 0000**
AND result (subnet number)	10.150.0.0	0000 1010 0110 001**0 0000 0000 0000 0000**
Change host to 1s (broadcast address)	10.151.255.255	0000 1010 0110 011**1 1111 1111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.150.0.1 through 10.151.255.254

Table A-18 lists the way to get the same answers using the subnet chart and magic math described in Chapter 12. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table A-18 *Question 6: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	10	150	200	200
Mask	255	254	0	0
Subnet number	10	150	0	0
First valid address	10	150	0	1
Broadcast	10	151	255	255
Last valid address	10	151	255	254

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 254 = 2 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 150 is the multiple of 2 that's closest to 150 but not bigger than 150. So, the second octet of the subnet number is 150.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 12, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 150 + 2 – 1 = 151.

Question 7: Answer

Table A-19 *Question 7: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	10.220.100.18	N/A
Mask	255.255.0.0	N/A
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	16	Always defined as number of binary 0s in mask
Number of subnet bits	8	32 – (network size + host size)
Number of subnets	$2^8 - 2 = 254$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{16} - 2 = 65,534$	$2^{\text{number-of-host-bits}} - 2$

The binary calculations of the subnet number and broadcast address are in Table A-20. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table A-20 *Question 7: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.220.100.18	0000 1010 1101 1100 **0110 0100 0001 0010**
Mask	255.255.0.0	1111 1111 1111 1111 **0000 0000 0000 0000**
AND result (subnet number)	10.220.0.0	0000 1010 1101 1100 **0000 0000 0000 0000**
Change host to 1s (broadcast address)	10.220.255.255	0000 1010 1101 1100 **1111 1111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.220.0.1 through 10.220.255.254

Table A-21 lists the way to get the same answers using the subnet chart and magic math described in Chapter 12.

Table A-21 *Question 7: Subnet, Broadcast, First, and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	10	220	100	18
Mask	255	255	0	0
Subnet number	10	220	0	0
First valid address	10	220	0	1
Broadcast	10	220	255	255
Last valid address	10	220	255	254

This subnetting scheme uses an easy mask because all of the octets are a 0 or a 255. No math tricks are needed at all!

Question 8: Answer

Table A-22 *Question 8: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	**Example**	**Rules to Remember**
Address	10.220.100.18	N/A
Mask	255.255.128.0	N/A
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	15	Always defined as number of binary 0s in mask
Number of subnet bits	9	32 – (network size + host size)
Number of subnets	$2^9 - 2 = 510$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{15} - 2 = 32,766$	$2^{\text{number-of-host-bits}} - 2$

The binary calculations of the subnet number and broadcast address are in Table A-23. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table A-23 *Question 8: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.220.100.18	0000 1010 1101 1100 0**110 0100 0001 0010**
Mask	255.255.128.0	1111 1111 1111 1111 1**000 0000 0000 0000**
AND result (subnet number)	10.220.0.0	0000 1010 1101 1100 0**000 0000 0000 0000**
Change host to 1s (broadcast address)	10.220.127.255	0000 1010 1101 1100 0**111 1111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.220.0.1 through 10.220.127.254

Table A-24 lists the way to get the same answers using the subnet chart and magic math described in Chapter 12. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table A-24 *Question 8: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	10	220	100	18
Mask	255	255	128	0
Subnet number	10	220	0	0
First address	10	220	0	1
Broadcast	10	220	127	255
Last Adress	10	220	127	254

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 128 = 128 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 0 is the multiple of 128 that's closest to 100 but not bigger than 100. So, the third octet of the subnet number is 0.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 12, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 0 + 128 – 1 = 127.

This example tends to confuse people because a mask with 128 in it gives you subnet numbers that just do not seem to look right. Table A-25 gives you the answers for the first several subnets, just to make sure that you are clear about the subnets when using this mask with a Class A network.

Table A-25 *Question 8: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Zero Subnet	First Valid Subnet	Second Valid Subnet	Third Valid Subnet
Subnet	10.0.0.0	10.0.128.0	10.1.0.0	10.1.128.0
First address	10.0.0.1	10.0.128.1	10.1.0.1	10.1.128.1
Last address	10.0.127.254	10.0.255.254	10.1.127.254	10.1.255.254
Broadcast	10.0.127.255	10.0.255.255	10.1.127.255	10.1.255.255

Question 9: Answer

Table A-26 *Question 9: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	**Example**	**Rules to Remember**
Address	172.31.100.100	N/A
Mask	255.255.192.0	N/A
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	14	Always defined as number of binary 0s in mask
Number of subnet bits	2	32 – (network size + host size)
Number of subnets	$2^2 - 2 = 2$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{14} - 2 = 16{,}382$	$2^{\text{number-of-host-bits}} - 2$

The binary calculations of the subnet number and broadcast address are in Table A-27. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table A-27 *Question 9: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.31.100.100	1010 1100 0001 1111 01**10 0100 0110 0100**
Mask	255.255.192.0	1111 1111 1111 1111 11**00 0000 0000 0000**
AND result (subnet number)	172.31.64.0	1010 1100 0001 1111 01**00 0000 0000 0000**
Change host to 1s (broadcast address)	172.31.127.255	1010 1100 0001 1111 01**11 1111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.64.1 through 172.31.127.254

Table A-28 lists the way to get the same answers using the subnet chart and magic math described in Chapter 12. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table A-28 *Question 9: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	172	31	100	100
Mask	255	255	192	0
Subnet number	172	31	64	0
First valid address	172	31	64	1
Broadcast	172	31	127	255
Last valid address	172	31	127	254

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 192 = 64 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 64 is the multiple of 64 that's closest to 100 but not bigger than 100. So, the third octet of the subnet number is 64.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 12, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 64 + 64 – 1 = 127.

Question 10: Answer

Table A-29 *Question 10: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	172.31.100.100	N/A
Mask	255.255.224.0	N/A
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	13	Always defined as number of binary 0s in mask
Number of subnet bits	3	32 – (network size + host size)
Number of subnets	$2^3 - 2 = 6$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{13} - 2 = 8190$	$2^{\text{number-of-host-bits}} - 2$

The binary calculations of the subnet number and broadcast address are in Table A-30. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table A-30 *Question 10: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.31.100.100	1010 1100 0001 1111 011**0 0100 0110 0100**
Mask	255.255.224.0	1111 1111 1111 1111 111**0 0000 0000 0000**
AND result (subnet number)	172.31.96.0	1010 1100 0001 1111 011**0 0000 0000 0000**
Change host to 1s (broadcast address)	172.31.127.255	1010 1100 0001 1111 011**1 1111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.96.1 through 172.31.127.254

Table A-31 lists the way to get the same answers using the subnet chart and magic math described in Chapter 12. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table A-31 *Question 10: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	172	31	100	100
Mask	255	255	224	0
Subnet number	172	31	96	0
First valid address	172	31	96	1
Broadcast	172	31	127	255
Last valid address	172	31	127	254

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 224 = 32 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 96 is the multiple of 32 that's closest to 100 but not bigger than 100. So, the third octet of the subnet number is 96.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 12, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 96 + 32 – 1 = 127.

Question 11: Answer

Table A-32 *Question 11: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	**Example**	**Rules to Remember**
Address	172.31.200.10	N/A
Mask	255.255.240.0	N/A
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	12	Always defined as number of binary 0s in mask
Number of subnet bits	4	32 – (network size + host size)
Number of subnets	$2^4 - 2 = 14$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{12} - 2 = 4094$	$2^{\text{number-of-host-bits}} - 2$

Table A-33 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in bold print in the table.

Table A-33 *Question 11: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.31.200.10	1010 1100 0001 1111 1100 **1000 0000 1010**
Mask	255.255.240.0	1111 1111 1111 1111 1111 **0000 0000 0000**
AND result (subnet number)	172.31.192.0	1010 1100 0001 1111 1100 **0000 0000 0000**
Change host to 1s (broadcast address)	172.31.207.255	1010 1100 0001 1111 1100 **1111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.192.1 through 172.31.207.254

Table A-34 lists the way to get the same answers using the subnet chart and magic math described in Chapter 12. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table A-34 *Question 13: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	172	31	200	10
Mask	255	255	240	0
Subnet number	172	31	192	0
First valid address	172	31	192	1
Broadcast	172	31	207	255
Last valid address	172	31	207	254

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 240 = 16 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 192 is the multiple of 16 that's closest to 200 but not bigger than 200. So, the third octet of the subnet number is 192.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 12, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 192 + 16 – 1 = 207.

Question 12: Answer

Table A-35 *Question 12: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	**Example**	**Rules to Remember**
Address	172.31.200.10	N/A
Mask	255.255.248.0	N/A
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	11	Always defined as number of binary 0s in mask
Number of subnet bits	5	32 – (network size + host size)
Number of subnets	$2^5 - 2 = 30$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{11} - 2 = 2046$	$2^{\text{number-of-host-bits}} - 2$

Table A-36 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table A-36 *Question 12: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.31.200.10	1010 1100 0001 1111 1100 1**000 0000 1010**
Mask	255.255.248.0	1111 1111 1111 1111 1111 1**000 0000 0000**
AND result (subnet number)	172.31.200.0	1010 1100 0001 1111 1100 1**000 0000 0000**
Change host to 1s (broadcast address)	172.31.207.255	1010 1100 0001 1111 1100 1**111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.200.1 through 172.31.207.254

Table A-37 lists the way to get the same answers using the subnet chart and magic math described in Chapter 12. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table A-37 *Question 12: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	172	31	200	10
Mask	255	255	248	0
Subnet number	172	31	200	0
First valid address	172	31	200	1
Broadcast	172	31	207	255
Last valid address	172	31	207	254

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 248 = 8 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 200 is the multiple of 8 that's closest to 200 but not bigger than 200. So, the third octet of the subnet number is 200.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 12, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 200 + 8 – 1 = 207.

Question 13: Answer

Table A-38 *Question 13: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	172.31.50.50	N/A
Mask	255.255.252.0	N/A
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	10	Always defined as number of binary 0s in mask
Number of subnet bits	6	32 – (network size + host size)
Number of subnets	$2^6 - 2 = 62$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{10} - 2 = 1022$	$2^{\text{number-of-host-bits}} - 2$

Table A-39 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table A-39 *Question 13: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.31.50.50	1010 1100 0001 1111 0011 00**10 0011 0010**
Mask	255.255.252.0	1111 1111 1111 1111 1111 11**00 0000 0000**
AND result (subnet number)	172.31.48.0	1010 1100 0001 1111 0011 00**00 0000 0000**
Change host to 1s (broadcast address)	172.31.51.255	1010 1100 0001 1111 0011 00**11 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.48.1 through 172.31.51.254

Table A-40 lists the way to get the same answers using the subnet chart and magic math described in Chapter 12. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table A-40 *Question 13: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	**Octet 2**	**Octet 3**	**Octet 4**
Address	172	31	50	50
Mask	255	255	252	0
Subnet number	172	31	48	0
First valid address	172	31	48	1
Broadcast	172	31	51	255
Last valid address	172	31	51	254

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 252 = 4 in this case (256 –

mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 48 is the multiple of 4 that's closest to 50 but not bigger than 50. So, the third octet of the subnet number is 48.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 12, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 48 + 4 – 1 = 51.

Question 14: Answer

Table A-41 *Question 14: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	172.31.50.50	N/A
Mask	255.255.254.0	N/A
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	9	Always defined as number of binary 0s in mask
Number of subnet bits	7	32 – (network size + host size)
Number of subnets	$2^7 - 2 = 126$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^9 - 2 = 510$	$2^{\text{number-of-host-bits}} - 2$

Table A-42 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in bold print in the table.

Table A-42 *Question 14: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.31.50.50	1010 1100 0001 1111 0011 001**0 0011 0010**
Mask	255.255.254.0	1111 1111 1111 1111 1111 111**0 0000 0000**
AND result (subnet number)	172.31.50.0	1010 1100 0001 1111 0011 001**0 0000 0000**
Change host to 1s (broadcast address)	172.31.51.255	1010 1100 0001 1111 0011 001**1 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.50.1 through 172.31.51.254

Table A-43 lists the way to get the same answers using the subnet chart and magic math described in Chapter 12. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table A-43 *Question 14: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	172	31	50	50
Mask	255	255	254	0
Subnet number	172	31	50	0
First valid address	172	31	50	1
Broadcast	172	31	51	255
Last valid address	172	31	51	254

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 254 = 2 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 50 is the multiple of 2 that's closest to 50 but not bigger than 50. So, the third octet of the subnet number is 50.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 12, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 50 + 2 – 1 = 51.

Question 15: Answer

Table A-44 *Question 15: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	**Example**	**Rules to Remember**
Address	172.31.140.14	N/A
Mask	255.255.255.0	N/A
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	8	Always defined as number of binary 0s in mask
Number of subnet bits	8	32 – (network size + host size)
Number of subnets	$2^8 - 2 = 254$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^8 - 2 = 254$	$2^{\text{number-of-host-bits}} - 2$

Table A-45 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table A-45 *Question 15: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.31.140.14	1010 1100 0001 1111 1000 1100 **0000 1110**
Mask	255.255.255.0	1111 1111 1111 1111 1111 1111 **0000 0000**
AND result (subnet number)	172.31.140.0	1010 1100 0001 1111 1000 1100 **0000 0000**
Change host to 1s (broadcast address)	172.31.140.255	1010 1100 0001 1111 1000 1100 **1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.140.1 through 172.31.140.254

Table A-46 lists the way to get the same answers using the subnet chart and magic math described in Chapter 12.

Table A-46 *Question 15: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	172	31	140	14
Mask	255	255	255	0
Subnet number	172	31	140	0
First valid address	172	31	140	1
Broadcast	172	31	140	255
Last valid address	172	31	140	254

This subnetting scheme uses an easy mask because all of the octets are a 0 or a 255. No math tricks are needed at all!

Question 16: Answer

Table A-47 *Question 16: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	172.31.140.14	N/A
Mask	255.255.255.128	N/A
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	7	Always defined as number of binary 0s in mask
Number of subnet bits	9	32 – (network size + host size)
Number of subnets	$2^9 - 2 = 510$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^7 - 2 = 126$	$2^{\text{number-of-host-bits}} - 2$

Table A-48 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table A-48 *Question 16: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.31.140.14	1010 1100 0001 1111 1000 1100 0**000 1110**
Mask	255.255.255.128	1111 1111 1111 1111 1111 1111 1**000 0000**
AND result (subnet number)	172.31.140.0	1010 1100 0001 1111 1000 1100 0**000 0000**
Change host to 1s (broadcast address)	172.31.140.127	1010 1100 0001 1111 1000 1100 0**111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.140.1 through 172.31.140.126

Table A-49 lists the way to get the same answers using the subnet chart and magic math described in Chapter 12. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table A-49 *Question 16: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	172	31	140	14
Mask	255	255	255	128
Subnet number	172	31	140	0
First valid address	172	31	140	1
Broadcast	172	31	140	127
Last valid address	172	31	140	126

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The fourth octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 128 = 128 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 0 is the multiple of 128 that's closest to 14 but not bigger than 14. So, the fourth octet of the subnet number is 0.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 12, but the tricky part is, as usual, in the "interesting" octet.

Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 0 + 128 – 1 = 127.

Question 17: Answer

Table A-50 *Question 17: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	192.168.15.150	N/A
Mask	255.255.255.192	N/A
Number of network bits	24	Always defined by Class A, B, C
Number of host bits	6	Always defined as number of binary 0s in mask
Number of subnet bits	2	32 – (network size + host size)
Number of subnets	$2^2 - 2 = 2$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^6 - 2 = 62$	$2^{\text{number-of-host-bits}} - 2$

Table A-51 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table A-51 *Question 17: Binary Calculation of Subnet and Broadcast Addresses*

Address	192.168.15.150	1100 0000 1010 1000 0000 1111 10**01 0110**
Mask	255.255.255.192	1111 1111 1111 1111 1111 1111 11**00 0000**
AND result (subnet number)	192.168.15.128	1100 0000 1010 1000 0000 1111 10**00 0000**
Change host to 1s (broadcast address)	192.168.15.191	1100 0000 1010 1000 0000 1111 10**11 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

192.168.15.129 through 192.168.15.190

Table A-52 lists the way to get the same answers using the subnet chart and magic math described in Chapter 12. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table A-52 *Question 17: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	192	168	15	150
Mask	255	255	255	192
Subnet number	192	168	15	128
First valid address	192	168	15	129
Broadcast	192	168	15	191
Last valid address	192	168	15	190

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The fourth octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 192 = 64 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 128 is the multiple of 64 that's closest to 150 but not bigger than 150. So, the fourth octet of the subnet number is 128.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 12, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 128 + 64 – 1 = 191.

Question 18: Answer

Table A-53 *Question 18: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	192.168.15.150	N/A
Mask	255.255.255.224	N/A
Number of network bits	24	Always defined by Class A, B, C
Number of host bits	5	Always defined as number of binary 0s in mask
Number of subnet bits	3	32 – (network size + host size)
Number of subnets	$2^3 - 2 = 6$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^5 - 2 = 30$	$2^{\text{number-of-host-bits}} - 2$

Table A-54 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table A-54 *Question 18: Binary Calculation of Subnet and Broadcast Addresses*

Address	192.168.15.150	1100 0000 1010 1000 0000 1111 100**1 0110**
Mask	255.255.255.224	1111 1111 1111 1111 1111 1111 111**0 0000**
AND result (subnet number)	192.168.15.128	1100 0000 1010 1000 0000 1111 100**0 0000**
Change host to 1s (broadcast address)	192.168.15.159	1100 0000 1010 1000 0000 1111 100**1 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

192.168.15.129 through 192.168.15.158

Table A-55 lists the way to get the same answers using the subnet chart and magic math described in Chapter 12. Remember, subtracting the interesting (non-0 or 255) mask

value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table A-55 *Question 18: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	192	168	15	150
Mask	255	255	255	224
Subnet number	192	168	15	128
First valid address	192	168	15	129
Broadcast	192	168	15	159
Last valid address	192	168	15	158

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The fourth octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 224 = 32 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 128 is the multiple of 32 that's closest to 150 but not bigger than 150. So, the fourth octet of the subnet number is 128.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 12, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 128 + 32 – 1 = 159.

Question 19: Answer

Table A-56 *Question 19: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	192.168.100.100	N/A
Mask	255.255.255.240	N/A
Number of network bits	24	Always defined by Class A, B, C
Number of host bits	4	Always defined as number of binary 0s in mask
Number of subnet bits	4	32 – (network size + host size)
Number of subnets	$2^4 - 2 = 14$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^4 - 2 = 14$	$2^{\text{number-of-host-bits}} - 2$

Table A-57 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table A-57 *Question 19: Binary Calculation of Subnet and Broadcast Addresses*

Address	192.168.100.100	1100 0000 1010 1000 0110 0100 0110 **0100**
Mask	255.255.255.240	1111 1111 1111 1111 1111 1111 1111 **0000**
AND result (subnet number)	192.168.100.96	1100 0000 1010 1000 0110 0100 0110 **0000**
Change host to 1s (broadcast address)	192.168.100.111	1100 0000 1010 1000 0110 0100 0110 **1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

192.168.100.97 through 192.168.100.110

Table A-58 lists the way to get the same answers using the subnet chart and magic math described in Chapter 12. Remember, subtracting the interesting (non-0 or 255) mask

value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table A-58 *Question 19: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	192	168	100	100
Mask	255	255	255	240
Subnet number	192	168	100	96
First valid address	192	168	100	97
Broadcast	192	168	100	111
Last valid address	192	168	100	110

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The fourth octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 240 = 16 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 96 is the multiple of 16 that's closest to 100 but not bigger than 100. So, the fourth octet of the subnet number is 96.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 12, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 96 + 16 – 1 = 111.

Question 20: Answer

Table A-59 *Question 20: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	192.168.100.100	N/A
Mask	255.255.255.248	N/A
Number of network bits	24	Always defined by Class A, B, C
Number of host bits	3	Always defined as number of binary 0s in mask
Number of subnet bits	5	32 – (network size + host size)
Number of subnets	$2^5 - 2 = 30$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^3 - 2 = 6$	$2^{\text{number-of-host-bits}} - 2$

Table A-60 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table A-60 *Question 20: Binary Calculation of Subnet and Broadcast Addresses*

Address	192.168.100.100	1100 0000 1010 1000 0110 0100 0110 0**100**
Mask	255.255.255.248	1111 1111 1111 1111 1111 1111 1111 1**000**
AND result (subnet number)	192.168.100.96	1100 0000 1010 1000 0110 0100 0110 0**000**
Change host to 1s (broadcast address)	192.168.100.103	1100 0000 1010 1000 0110 0100 0110 0**111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

192.168.100.97 through 192.168.100.102

Table A-61 lists the way to get the same answers using the subnet chart and magic math described in Chapter 12. Remember, subtracting the interesting (non-0 or 255) mask

value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table A-61 *Question 20: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	192	168	100	100
Mask	255	255	255	248
Subnet number	192	168	100	96
First valid address	192	168	100	97
Broadcast	192	168	100	103
Last valid address	192	168	100	102

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The fourth octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 248 = 8 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 96 is the multiple of 8 that's closest to 100 but not bigger than 100. So, the fourth octet of the subnet number is 96.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 12, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 96 + 8 – 1 = 103.

Question 21: Answer

Table A-62 *Question 21: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	**Example**	**Rules to Remember**
Address	192.168.15.230	N/A
Mask	255.255.255.252	N/A
Number of network bits	24	Always defined by Class A, B, C
Number of host bits	2	Always defined as number of binary 0s in mask
Number of subnet bits	6	32 – (network size + host size)
Number of subnets	$2^6 - 2 = 62$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^2 - 2 = 2$	$2^{\text{number-of-host-bits}} - 2$

Table A-63 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table A-63 *Question 21: Binary Calculation of Subnet and Broadcast Addresses*

Address	192.168.15.230	1100 0000 1010 1000 0000 1111 1110 01**10**
Mask	255.255.255.252	1111 1111 1111 1111 1111 1111 1111 11**00**
AND result (subnet number)	192.168.15.228	1100 0000 1010 1000 0000 1111 1110 01**00**
Change host to 1s (broadcast address)	192.168.15.231	1100 0000 1010 1000 0000 1111 1110 01**11**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

192.168.15.229 through 192.168.15.230

Table A-64 lists the way to get the same answers using the subnet chart and magic math described in Chapter 12. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table A-64 *Question 21: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	192	168	15	230
Mask	255	255	255	252
Subnet number	192	168	15	228
First valid address	192	168	15	229
Broadcast	192	168	15	231
Last valid address	192	168	15	230

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The fourth octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 252 = 4 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 228 is the multiple of 4 that's closest to 230 but not bigger than 230. So, the fourth octet of the subnet number is 228.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 12, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 228 + 4 – 1 = 231.

Question 22: Answer

Table A-65 *Question 22: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	**Example**	**Rules to Remember**
Address	10.1.1.1	N/A
Mask	255.248.0.0	N/A
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	19	Always defined as number of binary 0s in mask
Number of subnet bits	5	32 – (network size + host size)
Number of subnets	$2^5 - 2 = 30$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{19} - 2 = 524,286$	$2^{\text{number-of-host-bits}} - 2$

Table A-66 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table A-66 *Question 22: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.1.1.1	0000 1010 0000 0**001 0000 0001 0000 0001**
Mask	255.248.0.0	1111 1111 1111 1**000 0000 0000 0000 0000**
AND result (subnet number)	10.0.0.0	0000 1010 0000 0**000 0000 0000 0000 0000**
Change host to 1s (broadcast address)	10.7.255.255	0000 1010 0000 0**111 1111 1111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.0.0.1 through 10.7.255.254

Take a closer look at the subnet part of the subnet address, as is shown in bold here: 0000 1010 **0000 0**000 0000 0000 0000 0000. The subnet part of the address is all binary 0s, making this subnet a zero subnet. This subnet should be avoided unless you are running out of available subnets to use.

Table A-67 lists the way to get the same answers using the subnet chart and magic math described in Chapter 12. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table A-67 *Question 22: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	10	1	1	1
Mask	255	248	0	0
Subnet number	10	0	0	0
First valid address	10	0	0	1
Broadcast	10	7	255	255
Last valid address	10	7	255	254

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 248 = 8 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 0 is the multiple of 8 that's closest to 1 but not bigger than 1. So, the second octet of the subnet number is 0.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 12, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 0 + 8 – 1 = 7.

Question 23: Answer

Table A-68 *Question 23: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	**Example**	**Rules to Remember**
Address	172.16.1.200	N/A
Mask	255.255.240.0	N/A
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	12	Always defined as number of binary 0s in mask
Number of subnet bits	4	32 – (network size + host size)
Number of subnets	$2^4 - 2 = 14$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{12} - 2 = 4094$	$2^{\text{number-of-host-bits}} - 2$

Table A-69 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table A-69 *Question 23: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.16.1.200	1010 1100 0001 0000 0000 **0001 1100 1000**
Mask	255.255.240.0	1111 1111 1111 1111 1111 **0000 0000 0000**
AND result (subnet number)	172.16.0.0	1010 1100 0001 0000 0000 **0000 0000 0000**
Change host to 1s (broadcast address)	172.16.15.255	1010 1100 0001 0000 0000 **1111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.16.0.1 through 172.16.15.254

Take a closer look at the subnet part of the subnet address, as shown in bold here: 1010 1100 0001 0000 **0000** 0000 0000 0000. The subnet part of the address is all binary 0s, making this subnet a zero subnet. This subnet should be avoided unless you are running out of available subnets to use.

Table A-70 lists the way to get the same answers using the subnet chart and magic math described in Chapter 12. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table A-70 *Question 23: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	172	16	1	200
Mask	255	255	240	0
Subnet number	172	16	0	0
First valid address	172	16	0	1
Broadcast	172	16	15	255
Last valid address	172	16	15	254

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 240 = 16 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 0 is the multiple of 16 that's closest to 1 but not bigger than 1. So, the third octet of the subnet number is 0.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 12, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 0 + 16 – 1 = 15.

Question 24: Answer

Table A-71 *Question 24: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	172.16.0.200	N/A
Mask	255.255.255.192	N/A
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	6	Always defined as number of binary 0s in mask
Number of subnet bits	10	32 – (network size + host size)
Number of subnets	$2^{10} - 2 = 1022$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{6} - 2 = 62$	$2^{\text{number-of-host-bits}} - 2$

Table A-72 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table A-72 *Question 24: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.16.0.200	1010 1100 0001 0000 0000 0000 11**00 1000**
Mask	255.255.255.192	1111 1111 1111 1111 1111 1111 11**00 0000**
AND result (subnet number)	172.16.0.192	1010 1100 0001 0000 0000 0000 11**00 0000**
Change host to 1s (broadcast address)	172.16.0.255	1010 1100 0001 0000 0000 0000 11**11 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.16.0.193 through 172.16.0.254

Table A-73 lists the way to get the same answers using the subnet chart and magic math described in Chapter 12. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table A-73 *Question 24: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	172	16	0	200
Mask	255	255	255	192
Subnet number	172	16	0	192
First valid address	172	16	0	193
Broadcast	172	16	0	255
Last valid address	172	16	0	254

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The fourth octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 192 = 64 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 192 is the multiple of 64 that's closest to 200 but not bigger than 200. So, the fourth octet of the subnet number is 192.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 12, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 192 + 64 – 1 = 255.

You can easily forget that the subnet part of this address, when using this mask, actually covers all of the third octet as well as 2 bits of the fourth octet. For instance, the valid subnet numbers in order are listed here, starting with the first valid subnet by avoiding subnet 172.16.0.0—the zero subnet in this case:

172.16.0.64
172.16.0.128
172.16.0.192
172.16.1.0
172.16.1.64
172.16.1.128
172.16.1.192
172.16.2.0
172.16.2.64
172.16.2.128
172.16.2.192
172.16.3.0
172.16.3.64
172.16.3.128
172.16.3.192

And so on.

Question 25: Answer

Congratulations, you made it through all the extra subnetting practice! Here's an easy one to complete your review—one with no subnetting at all!

Table A-74 *Question 25: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	10.1.1.1	N/A
Mask	255.0.0.0	N/A
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	24	Always defined as number of binary 0s in mask
Number of subnet bits	0	32 – (network size + host size)
Number of subnets	0	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{24} - 2 =$ 16,777,214	$2^{\text{number-of-host-bits}} - 2$

Table A-75 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in bold print in the table.

Table A-75 *Question 25: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.1.1.1	0000 1010 **0000 0001 0000 0001 0000 0001**
Mask	255.0.0.0	1111 1111 **0000 0000 0000 0000 0000 0000**
AND result (subnet number)	10.0.0.0	0000 1010 **0000 0000 0000 0000 0000 0000**
Change host to 1s (broadcast address)	10.255.255.255	0000 1010 **1111 1111 1111 1111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.0.0.1 through 10.255.255.254

Table A-76 lists the way to get the same answers using the subnet chart and magic math described in Chapter 12.

Table A-76 *Question 25: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	10	1	1	1
Mask	255	0	0	0
Network number	10	0	0	0
First valid address	10	0	0	1
Broadcast	10	255	255	255
Last valid address	10	255	255	254

APPENDIX B

Scenarios

This appendix contains a set of lab scenarios, each of which contains some type of problem statement that you need to solve. Some scenarios ask specific questions. Others ask that you configure the devices in a network. Some scenarios give you command output from a working network, and ask that you decipher the output to describe the current status in the network, or figure out a problem in the network. In any case, these scenarios help you exercise your knowledge of Cisco network designs, configuration commands, and show commands.

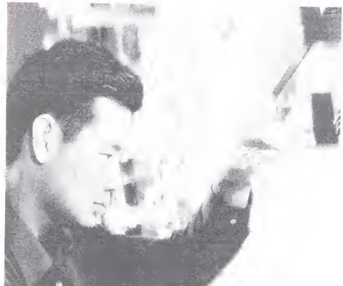
The scenarios revolve around the coverage in Chapters 7, 8, 9, and 12 in this book (the *CCNA INTRO Exam Certification Guide*). To keep them organized, the scenarios are numbered with the corresponding chapter number in the first part of the scenario number. Table B-1 lists the scenarios found in this appendix.

Table B-1 *List of Scenarios in this Chapter*

Scenario Number	Chapter which covers the topics	Description
1	7	Configuration Comparisons
2	7	More Configuration Comparisons
3*	8	LAN Switch Basic Configuration
4	9	Broadcast and Collision Domain Analysis
5	12	IP Addressing and Subnet Calculation
6*	12	Subnet Design with a Class B Network

*These labs' configurations can be performed using the special version of Boson's Netsim network simluator that comes with book CD. Refer to appendix C in the book (not the CD appendix C) for a complete list of all available hands-on exercises that can be done using Netsim.

You can either perform these scenarios as part of your final preparation for the INTRO exam, or use them at the end of each chapter. Regardless, just pick the scenarios you want to do, and dive in! For those of you wanting to do some of these scenarios using Boson NetSim, refer to appendix C in the book for more details on how to start NetSim.



Scenarios for Chapter 7

Scenario 1

Compare the following output in Example B-1 and Example B-2. Example B-1 was gathered at 11:00 a.m., 30 minutes earlier than in Example B-2. What can you definitively say happened to this router during the intervening half hour?

Example B-1 *11:00 a.m.* **show running-config**

```
hostname Gorno
!
enable password cisco
!
interface Serial0
 ip address 134.141.12.1 255.255.255.0
!
interface Serial1
 ip address 134.141.13.1 255.255.255.0
!
interface Ethernet0
 ip address 134.141.1.1 255.255.255.0
!
router rip
 network 134.141.0.0
!
line con 0
 password cisco
 login
line aux 0
line vty 0 4
 password cisco
 login
```

Example B-2 *11:30 a.m.* **show running-config**

```
hostname SouthernSiberia
prompt Gorno
!
enable secret $8df003j56ske92
enable password cisco
!
interface Serial0
 ip address 134.141.12.1 255.255.255.0
!
```

Example B-2 *11:30 a.m.* **show running-config** (Continued)

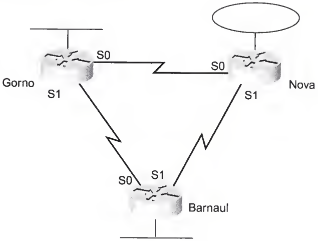
```
interface Serial1
 ip address 134.141.13.1 255.255.255.0
!
interface Ethernet0
 ip address 134.141.1.1 255.255.255.0
 no cdp enable
!
router rip
 network 134.141.0.0
!
line con 0
 password cisco
 login
line aux 0
line vty 0 4
 password cisco
 login
```

Questions on Scenario 1

1. During the process of changing the configuration in Scenario 1, the command prompt temporarily was SouthernSiberia(config)#. What configuration commands, and in what order, could have changed the configuration as shown and allowed the prompt to temporarily be SouthernSiberia(config)#?

2. Assuming that Figure B-1 is complete, what effect does the no cdp enable command have?

Figure B-1 *Siberian Enterprises' Sample Network*



3. What effect would the no enable password cisco command have at this point?

Scenario 2

Example B-3 shows that the show **running-config** command was executed on the Nova router.

Example B-3 *Configuration of Router Nova*

```
hostname Nova
banner # This is the router in Nova Sibiersk; Dress warmly before entering! #
!
boot system tftp c2500-js-113.bin 134.141.88.3
boot system flash c2500-j-l.111-9.bin
boot system rom
!
enable password cisco
!
interface Serial0
 ip address 134.141.12.2 255.255.255.0
!
interface Serial1
 ip address 134.141.23.2 255.255.255.0
!
interface TokenRing0
 ip address 134.141.2.2 255.255.255.0
!
router rip
 network 134.141.0.0
!
line con 0
 password cisco
 login
line aux 0
line vty 0 4
 password cisco
 login
```

Questions on Scenario 2

1. If this is all the information that you have, what IOS image do you expect will be loaded when the user reloads Nova?

2. Examine the following command output in Example B-4, taken immediately before the user is going to type the reload command. What IOS image do you expect will be loaded?

Example B-4 show ip route *Command Output for Nova*

```
Nova#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route


Gateway of last resort is not set

     134.141.0.0/24 is subnetted, 6 subnets
C       134.141.2.0 is directly connected, TokenRing0
R       134.141.3.0 [120/1] via 134.141.23.3, 00:00:15, Serial1
R       134.141.1.0 [120/1] via 134.141.12.1, 00:00:20, Serial0
C       134.141.12.0 is directly connected, Serial0
R       134.141.13.0 [120/1] via 134.141.12.1, 00:00:20, Serial0
                     [120/1] via 134.141.23.3, 00:00:15, Serial1
C       134.141.23.0 is directly connected, Serial1
```

3. Now examine the following **show flash** command in Example B-5, which was issued immediately after the **show ip route** command in Example B-4 but before the user issued the **reload** command. What IOS image do you think would be loaded in this case?

Example B-5 show flash *Command Output for Nova*

```
Nova#show flash
4096K bytes of flash memory sized on embedded flash.
File   name/status
 0 c2500-j-l.111-3.bin
[682680/4194304 bytes free/total]
```

4. Now examine the configuration in Example B-6. Assume that there is now a route to 134.141.88.0 and that the file *c2500-j-l.111-9.bin* is an IOS image in Flash memory. What IOS image do you expect will be loaded now?

Example B-6 show running-config *Command Output for Router Nova*

```
hostname Nova
banner # This is the router in Nova Sibiersk; Dress warmly before entering! #
!
boot system tftp c2500-js-113.bin 134.141.88.3
boot system flash c2500-j-l.111-9.bin
!
enable password cisco
```

continues

Example B-6 **show running-config** *Command Output for Router Nova (Continued)*

```
!
interface Serial0
 ip address 134.141.12.2 255.255.255.0
!
interface Serial1
 ip address 134.141.23.2 255.255.255.0
!
interface Ethernet0
 ip address 134.141.2.2 255.255.255.0
!
router rip
 network 134.141.0.0
!
line con 0
 password cisco
 login
line aux 0
line vty 0 4
 password cisco
 login
!
config-register 0x2101
```

Answers to Chapter 7 Scenarios

Scenario 1 Answers

In Scenario 1, the following commands were added to the configuration:

- **enable secret** as a global command.
- **prompt** as a global command.
- **no cdp enable** as an Ethernet0 subcommand.
- The **hostname** command also was changed.

The scenario questions' answers are as follows:

1. If the host name was changed to SouthSiberia first and the **prompt** command was added next, the prompt would have temporarily been SouthSiberia. Configuration commands are added to the RAM configuration file immediately and are used. In this case, when the **prompt** command was added, it caused the router to use Gorno, not the then-current host name SouthernSiberia, as the prompt.
2. No practical effect takes place. Because no other Cisco CDP–enabled devices are on that Ethernet, CDP messages from Gorno are useless. So, the only effect is to lessen the overhead on that Ethernet in a very small way.
3. No effect takes place, other than cleaning up the configuration file. The **enable** password is not used if an **enable** secret is configured.

Scenario 2 Answers

The answers to the questions in Scenario 2 are as follows:

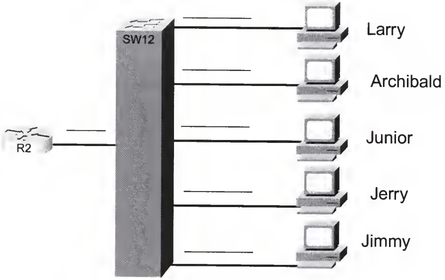
1. The first boot system statement would be used: **boot system tftp** c2500-js-113.bin 134.141.88.3.
2. The **boot system flash** command would be used. The TFTP boot presumably would fail because there is not currently a route to the subnet of which the TFTP server is a part. It is reasonable to assume that a route would not be learned 2 minutes later when the router had reloaded. So, the next **boot system** command (flash) would be used.
3. The **boot system ROM** command would be used. Because there is no file in Flash memory called c2500-j- l.111-9.bin, the boot from Flash memory would fail as well, leaving only one **boot** command.
4. IOS from ROM would be loaded because of the configuration register. If the configuration register boot field is set to 0x1, **boot system** commands are ignored. So, having a route to the 134.141.88.0/ 24 subnet and having *c2500-j-l.111-9.bin* in Flash memory does not help.

Scenario for Chapter 8

Scenario 3: LAN Switch Configuration

Your job is to deploy a new LAN switch at a remote site. Figure B-2 depicts the network. Perform the activities in the list that follows the diagram.

Figure B-2 *Scenario 3: Basic LAN Switch Configuration*



1. Clear the saved configuration before starting. Reload the switch so that it has no useful configuration.
2. Assign IP address 172.16.2.254, mask 255.255.255.0, to SW12. Assign it an appropriate default gateway, and configure SW12 to use a DNS, which is at 172.16.1.250.
3. Assign a host name of SW12.

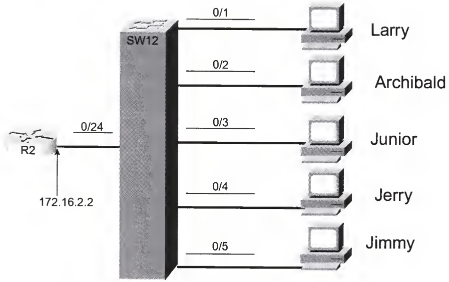
4. Choose port numbers to be used for each device, as if you were planning the physical installation. Write down these numbers on the diagram.
5. Configure so that the router uses 100-Mbps full-duplex operation.
6. Configure Archibald's MAC address so that it never leaves the address table.
7. List the commands that you would use to verify all of these features.
8. Identify the command that you would use to examine the running configuration, saved configuration, and IOS level.

Answers to Chapter 8 Scenario

Scenario 3 Answers

This scenario should have forced you to perform basic LAN configuration. Figure B-3 lists the port numbers used for the solution. Example B-7 lists the output from actually performing these steps sequentially on a 2950 series switch. An explanation of the steps follows the example.

Figure B-3 *Switch Port Numbers Used in Scenario 3 Answer*



Example B-7 *Scenario 3 Configuration and* show *Commands*

```
SW12#erase startup-config
Erasing the nvram filesystem will remove all files! Continue? [confirm]
[OK]
Erase of nvram: complete
SW12#reload
Proceed with reload? [confirm]

00:08:39: %SYS-5-RELOAD: Reload requestedBase ethernet MAC Address: 00:0a:b7:dc:
b7:80
Xmodem file system is available.
The password-recovery mechanism is enabled.
Initializing Flash...
```

Example B-7 *Scenario 3 Configuration and* **show** *Commands (Continued)*

```
flashfs[0]: 32 files, 6 directories
flashfs[0]: 0 orphaned files, 0 orphaned directories
flashfs[0]: Total bytes: 15998976
flashfs[0]: Bytes used: 9448960
flashfs[0]: Bytes available: 6550016
flashfs[0]: flashfs fsck took 17 seconds.
...done Initializing Flash.
Boot Sector Filesystem (bs:) installed, fsid: 3
Loading "flash:c3550-i5q3l2-mz.121-11.EA1"...flash:c3550-i5q3l2-mz.121-11.EA1: is a
directory

Error loading "flash:c3550-i5q3l2-mz.121-11.EA1"

Interrupt within 5 seconds to abort boot process.
Loading "flash:/c3550-i5q3l2-mz.121-11.EA1/c3550-i5q3l2-mz.121-
11.EA1.bin"...###################################################################################
##################################################################################################
##################################################################################################
##################################################################################################
###########################################################################################

File "flash:/c3550-i5q3l2-mz.121-11.EA1/c3550-i5q3l2-mz.121-11.EA1.bin" uncompre
ssed and installed, entry point: 0x3000
executing...

              Restricted Rights Legend

Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
 of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
 (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

           cisco Systems, Inc.
           170 West Tasman Drive
           San Jose, California 95134-1706



Cisco Internetwork Operating System Software
IOS (tm) C3550 Software (C3550-I5Q3L2-M), Version 12.1(11)EA1, RELEASE SOFTWARE
(fc1)
Copyright  1986-2002 by cisco Systems, Inc.
Compiled Wed 28-Aug-02 10:03 by antonino
Image text-base: 0x00003000, data-base: 0x0071D658
```

continues

Example B-7 *Scenario 3 Configuration and* show *Commands (Continued)*

```
Initializing flashfs...
flashfs[1]: 32 files, 6 directories
flashfs[1]: 0 orphaned files, 0 orphaned directories
flashfs[1]: Total bytes: 15998976
flashfs[1]: Bytes used: 9448960
flashfs[1]: Bytes available: 6550016
flashfs[1]: flashfs fsck took 8 seconds.
flashfs[1]: Initialization complete.
...done Initializing flashfs.
POST: CPU Buffer Tests : Begin
POST: CPU Buffer Tests : End, Status Passed
POST: CPU Interface Tests : Begin
POST: CPU Interface Tests : End, Status Passed
POST: Switch Core Tests : Begin
POST: Switch Core Tests : End, Status Passed
POST: CPU Interface 2nd Stage Tests : Begin
POST: CPU Interface 2nd Stage Tests : End, Status Passed
POST: CAM Subsystem Tests : Begin
POST: CAM Subsystem Tests : End, Status Passed
POST: Ethernet Controller Tests : Begin
POST: Ethernet Controller Tests : End, Status Passed
POST: Loopback Tests : Begin
POST: Loopback Tests : End, Status Passed

cisco WS-C3550-24 (PowerPC) processor (revision E0) with 65526K/8192K bytes of m
emory.
Processor board ID CHK0635W02H
Last reset from warm-reset
Bridging software.
Running Layer2/3 Switching Image

Ethernet-controller 1 has 12 Fast Ethernet/IEEE 802.3 interfaces

Ethernet-controller 2 has 12 Fast Ethernet/IEEE 802.3 interfaces

Ethernet-controller 3 has 1 Gigabit Ethernet/IEEE 802.3 interface

Ethernet-controller 4 has 1 Gigabit Ethernet/IEEE 802.3 interface

24 FastEthernet/IEEE 802.3 interface(s)
2 Gigabit Ethernet/IEEE 802.3 interface(s)

The password-recovery mechanism is enabled.
384K bytes of flash-simulated non-volatile configuration memory.
Base ethernet MAC Address: 00:0A:B7:DC:B7:80
Motherboard assembly number: 73-5700-08
Power supply part number: 34-0966-02
```

Example B-7 *Scenario 3 Configuration and* show *Commands (Continued)*

```
Motherboard serial number: CAT0634058Q
Power supply serial number: DCA06340P5K
Model revision number: E0
Motherboard revision number: D0
Model number: WS-C3550-24-SMI
System serial number: CHK0635W02H


Press RETURN to get started!



Switch>
Switch>enable
Switch#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#interface vlan 1
Switch(config-if)#ip address 172.16.2.254 255.255.255.0
Switch(config-if)#exit
Switch(config)#ip default-gateway 172.16.2.2
Switch(config)#ip name-server 172.16.1.250
Switch(config)#hostname SW12
SW12(config)#interface fastethernet 0/24
SW12(config-if)#duplex ?
  auto  Enable AUTO duplex configuration
  full  Force full duplex operation
  half  Force half-duplex operation

SW12(config-if)#duplex full
Duplex will not be set until speed is set to non-auto value
SW12(config-if)#speed 100
SW12(config-if)#duplex full
SW12(config-if)#interface fastethernet 0/1
SW12(config-if)#speed 100
SW12(config-if)#duplex auto
SW12(config-if)#
SW12(config-if)#^Z
SW12#show mac-address-table dynamic
          Mac Address Table
------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       ----        -----
   1    0000.0c4a.8bca    DYNAMIC     Fa0/2
   1    0007.8580.71b8    DYNAMIC     Fa0/4
   1    0007.8580.7208    DYNAMIC     Fa0/3
Total Mac Addresses for this criterion: 3
```

continues

Example B-7 *Scenario 3 Configuration and* show *Commands (Continued)*

```
SW12#show version
Cisco Internetwork Operating System Software
IOS (tm) C3550 Software (C3550-I5Q3L2-M), Version 12.1(11)EA1, RELEASE SOFTWARE
(fc1)
Copyright  1986-2002 by cisco Systems, Inc.
Compiled Wed 28-Aug-02 10:03 by antonino
Image text-base: 0x00003000, data-base: 0x0071D658

ROM: Bootstrap program is C3550 boot loader

SW12 uptime is 5 minutes
System returned to ROM by power-on
System image file is "flash:/c3550-i5q3l2-mz.121-11.EA1/c3550-i5q3l2-mz.121-11.E
A1.bin"

cisco WS-C3550-24 (PowerPC) processor (revision E0) with 65526K/8192K bytes of m
emory.
Processor board ID CHK0635W02H
Last reset from warm-reset
Bridging software.
Running Layer2/3 Switching Image

Ethernet-controller 1 has 12 Fast Ethernet/IEEE 802.3 interfaces

Ethernet-controller 2 has 12 Fast Ethernet/IEEE 802.3 interfaces

Ethernet-controller 3 has 1 Gigabit Ethernet/IEEE 802.3 interface

Ethernet-controller 4 has 1 Gigabit Ethernet/IEEE 802.3 interface

24 FastEthernet/IEEE 802.3 interface(s)
2 Gigabit Ethernet/IEEE 802.3 interface(s)

The password-recovery mechanism is enabled.
384K bytes of flash-simulated non-volatile configuration memory.
Base ethernet MAC Address: 00:0A:B7:DC:B7:80
Motherboard assembly number: 73-5700-08
Power supply part number: 34-0966-02
Motherboard serial number: CAT0634058Q
Power supply serial number: DCA06340P5K
Model revision number: E0
Motherboard revision number: D0
Model number: WS-C3550-24-SMI
System serial number: CHK0635W02H
Configuration register is 0x10F

SW12#show running-config
```

Example B-7 *Scenario 3 Configuration and* show *Commands (Continued)*

```
Building configuration...

Current configuration : 1538 bytes
!
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname SW12
!
!
ip subnet-zero
!
!
spanning-tree extend system-id
!
!
!
interface FastEthernet0/1
 no ip address
 speed 100
!
interface FastEthernet0/2
 no ip address
!
interface FastEthernet0/3
 no ip address
!
interface FastEthernet0/4
 no ip address
!
interface FastEthernet0/5
 no ip address
!
interface FastEthernet0/6
 no ip address
!
interface FastEthernet0/7
 no ip address
!
interface FastEthernet0/8
 no ip address
!
interface FastEthernet0/9
 no ip address
```

continues

Example B-7 *Scenario 3 Configuration and* show *Commands (Continued)*

```
!
interface FastEthernet0/10
 no ip address
!
interface FastEthernet0/11
 no ip address
!
interface FastEthernet0/12
 no ip address
!
interface FastEthernet0/13
 no ip address
!
interface FastEthernet0/14
 no ip address
!
interface FastEthernet0/15
 no ip address
!
interface FastEthernet0/16
 no ip address
!
interface FastEthernet0/17
 no ip address
!
interface FastEthernet0/18
 no ip address
!
interface FastEthernet0/19
 no ip address
!
interface FastEthernet0/20
 no ip address
!
interface FastEthernet0/21
 no ip address
!
interface FastEthernet0/22
 no ip address
!
interface FastEthernet0/23
 no ip address
!
interface FastEthernet0/24
 no ip address
 duplex full
 speed 100
```

Example B-7 *Scenario 3 Configuration and* show *Commands (Continued)*

```
!
interface GigabitEthernet0/1
 no ip address
!
interface GigabitEthernet0/2
 no ip address
!
interface Vlan1
 ip address 172.16.2.254 255.255.255.0
 shutdown
!
ip default-gateway 172.16.2.2
ip classless
ip http server
!
!
!
!
line con 0
line vty 5 15
!
end

SW12#show startup-config
%% Non-volatile configuration memory is not present
```

Example B-7 begins with the startup config being deleted using the erase startup-config configuration. The reload command reinitializes the switch, so that it has no configuration either in startup configuration or the running configuration.

The IP address of the switch is configured under interface vlan 1, but the default gateway is configured as a global command. The duplex settings, interestingly, can only be explicitly set if all autonegotiation is disabled by explicitly setting the speed as well. The error messages were left in the example just to show that point.

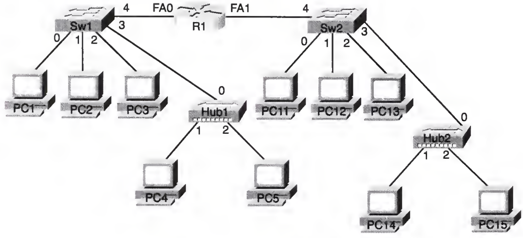
The rest of the commands in the example show the answers to the various questions. For instance, the show running-configuration command lists the running configuration, but the show startup-configuration command shows nothing, because the configuration has yet to be saved using the copy running-configuration startup-configuration command.

Scenario for Chapter 9

Scenario 4: LAN Switch Concepts

In this scenario, you will answer some questions about a simple network diagram. Figure B-4 depicts the network. Answer the questions that follow the diagram.

Figure B-4 *Scenario 4: Basic LAN Switch Concepts*



1. How many collision domains exist in this network?
2. How many broadcast domains exist in this network?
3. Assuming that all cards, switches, and router interfaces are 10/100 cards, how many ports total on each switch could run full duplex?
4. Assuming that all cards, switches, and router interfaces are 10/100 cards, how many ports total on each switch could run 100 Mbps?
5. The first frames to flow in this network are the following: PC5 sends an IP ARP, encapsulated in an Ethernet frame, looking for its default gateway, which is R1's FA0 interface's IP address. The Ethernet frame containing the ARP reply is the second frame.

Describe what ports each frame is sent out. Use Table B-2 to list where the frame flowed, or is draw on the diagram. If you use the table, write "received" if the frame was received in that port, or write "sent" if the frame was sent out that port.

Table B-2 *List of Hub/Switch/Router Ports for Figure B-4*

Port	Was Frame 1 Either Received in or Sent out This Port?	Was Frame 2 Either Received in or Sent out This Port?
Hub1port 0		
Hub1port 1		
Hub1port 2		
SW1port 0		
SW1port 1		
SW1port 2		
SW1port 3		
SW1port 4		
Hub2port 0		
Hub2port 1		
Hub2port 2		
SW2port 0		
SW2port 1		
SW2port 2		
SW2port 3		
SW2port 4		
R1FA0		
R1FA1		

Answers to Chapter 9 Scenario

Answers to Scenario 4

This scenario tests your recollection of a few of the core concepts for LAN switching. The answers are listed in succession:

1. Ten collision domains exist in the network for this scenario. Routers and switches separate LANs into separate collision domains, but shared hubs do not. In this diagram, each switch port and the device(s) connected to it form the individual collision domains.
2. Two broadcast domains exist in this network. Switches and hubs do not separate the LAN segments into different broadcast domains, but routers do. The two broadcast domains consist of the devices to the left of R1 and the devices to the right of R1.
3. Eight total switch ports could run full-duplex operation. Port 3 on each switch could not because there is a shared hub attached to this port, so collisions could happen. When collisions could happen, FDX is not allowed.
4. All ten switch ports could run 100 Mbps Fast Ethernet. Router FastEthernet interfaces support 100 Mbps, and the assumption was made that all the PCs support 100 Mbps. Shared hubs also can support 100 Mbps. So, all switch ports could run at 100 Mbps, but port 3 on each switch could not use full-duplex operation.
5. Figures B-5 and B-6 depict the flows of frame 1 and frame 2. Frame 1 has a source Ethernet address of PC5 and a broadcast destination address. Frame 2 has a source of R1's FA0 MAC address and a destination of PC5's MAC address. Table B-3 also describes the ports that the frames came in and out on their journeys, respectively.

Figure B-5 *Scenario 4: Path of First Frame in Question Number 5*

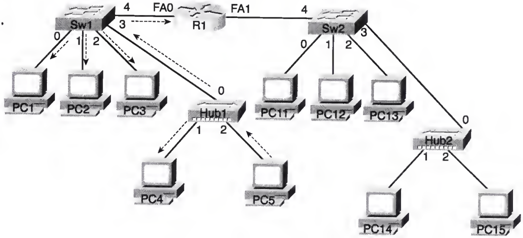


Figure B-6 *Scenario 4: Path of Second Frame in Question Number 5*

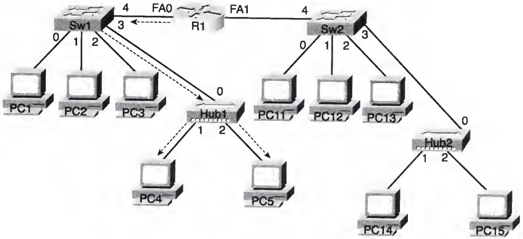


Table B-3 *Table of Incoming and Outgoing Ports for Frames in Scenario 4, Question 5*

Port	Was Frame 1 Either Received in or Sent out This Port?	Was Frame 2 Either Received in or Sent out This Port?
Hub1port 0	Sent	Received
Hub1port 1	Sent	Sent
Hub1port 2	Received	Sent
SW1port 0	Sent	

continues

Table B-3 *Table of Incoming and Outgoing Ports for Frames in Scenario 4, Question 5 (Continued)*

Port	Was Frame 1 Either Received in or Sent out This Port?	Was Frame 2 Either Received in or Sent out This Port?
SW1port 1	Sent	
SW1port 2	Sent	
SW1port 3	Received	Sent
SW1port 4	Sent	Received
Hub2port 0		
Hub2port 1		
Hub2port 2		
SW2port 0		
SW2port 1		
SW2port 2		
SW2port 3		
SW2port 4		
R1FA0	Received	Sent
R1FA1		

Frame 1 is a broadcast, so it must flow throughout the broadcast domain. So, Hub1 and Switch1 forward out all ports. R1, however, is the boundary of the broadcast domain, so R1 does not forward the broadcast. R1 replies to the ARP and encapsulates it in an Ethernet frame. This second frame has a destination of PC5's MAC address. SW1 learned that PC5's MAC is out its port 3. The hub did not learn anything because it does not keep an address table. So, R1 sends the second frame to PC5. SW1 forwards only out port 3, according to its address table. The hub still forwards out all ports.

Scenarios for Chapter 12

Scenario 5: IP Addressing and Subnet Calculation

Assume that you just took a new job. No one trusts you yet, so they will not give you any passwords to the router. Your mentor at your new company has left you at his desk while he goes to a meeting. He has left up a Telnet window, logged in to one router in user mode. In other words, you can issue only user-mode commands.

Assuming that you had issued the following commands (see Example B-8), draw the most specific network diagram that you can. Write the subnet numbers used on each link onto the diagram.

Example B-8 *Command Output on Router Fred*

```
fred>show interface
Serial0 is up, line protocol is up
  Hardware is HD64570
  Internet address is 199.1.1.65/27
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/0/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 1158 kilobits/sec

  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     27 packets input, 2452 bytes, 0 no buffer
     Received 27 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     29 packets output, 2044 bytes, 0 underruns
     0 output errors, 0 collisions, 28 interface resets
     0 output buffer failures, 0 output buffers swapped out
     7 carrier transitions
     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
Serial1 is up, line protocol is up
  Hardware is HD64570
```

continues

Example B-8 *Command Output on Router Fred (Continued)*

```
  Internet address is 199.1.1.97/27
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/0/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 1158 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     125 packets input, 7634 bytes, 0 no buffer
     Received 124 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     161 packets output, 9575 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 output buffer failures, 0 output buffers swapped out
     4 carrier transitions
     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
Ethernet0 is up, line protocol is up
Hardware is MCI Ethernet, address is 0000.0c55.AB44 (bia 0000.0c55.AB44)
  Internet address is 199.1.1.33/27
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output 00:00:00, output hang never
     Output queue 0/40, 0 drops; input queue 0/75, 0 drops
     Five minute input rate 4000 bits/sec, 4 packets/sec
     Five minute output rate 6000 bits/sec, 6 packets/sec
           22197 packets input, 309992 bytes, 0 no buffer
           Received 2343 broadcasts, 0 runts, 0 giants
           0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
           4456 packets output, 145765 bytes, 0 underruns
           3 output errors, 10 collisions, 2 interface resets, 0 restarts


fred>show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
```

Example B-8 *Command Output on Router Fred (Continued)*

```
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     199.1.1.0/27 is subnetted, 6 subnets
R       199.1.1.192 [120/1] via 199.1.1.98, 00:00:01, Serial1
R       199.1.1.128 [120/1] via 199.1.1.98, 00:00:01, Serial1
                    [120/1] via 199.1.1.66, 00:00:20, Serial0
R       199.1.1.160 [120/1] via 199.1.1.66, 00:00:20, Serial0
C       199.1.1.64 is directly connected, Serial0
C       199.1.1.96 is directly connected, Serial1
C       199.1.1.32 is directly connected, Ethernet0

fred>show ip protocol
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 23 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
 Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Redistributing: rip
  Default version control: send version 1, receive any version
    Interface        Send  Recv   Key-chain
    Serial0          1     1 2
    Serial1          1     1 2
    Ethernet0        1     1 2
  Automatic network summarization is in effect
  Maximum path: 4
 Routing for Networks:

    199.1.1.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    199.1.1.66           120      00:00:04
    199.1.1.98           120      00:00:14
  Distance: (default is 120)

fred>show cdp neighbor detail
-------------------------
Device ID: dino
Entry address(es):
  IP address: 199.1.1.66
Platform: Cisco 2500,  Capabilities: Router
Interface: Serial0,  Port ID (outgoing port): Serial0
Holdtime : 148 sec
```

continues

Example B-8 *Command Output on Router Fred (Continued)*

```
Version :
Cisco Internetwork Operating System Software
IOS (tm) 2500 Software (C2500-DS-L), Version 12.2(1), RELEASE SOFTWARE (fc2)
Copyright  1986-2001 by cisco Systems, Inc.
Compiled Fri 27-Apr-01 14:43 by cmong

advertisement version: 2
-------------------------
Device ID: Barney
Entry address(es):
  IP address: 199.1.1.98
Platform: Cisco 2500,  Capabilities: Router
Interface: Serial1,  Port ID (outgoing port): Serial0
Holdtime : 155 sec

Version :
Cisco Internetwork Operating System Software
IOS (tm) 2500 Software (C2500-DS-L), Version 12.2(1), RELEASE SOFTWARE (fc2)
Copyright  1986-2001 by cisco Systems, Inc.
Compiled Fri 27-Apr-01 14:43 by cmong

advertisement version: 2
```

Scenario 6: IP Subnet Design with a Class B Network

Your job is to plan a new network. The topology required includes three sites, one Ethernet at each site, and point-to-point serial links for connectivity, as shown in Figure B-7. The network might grow to need at most 100 subnets, with 200 hosts per subnet maximum. Use network 172.16.0.0, and use the same subnet mask for all subnets. Use Table B-4 to record your choices, or use a separate piece of paper.

Figure B-7 *Scenario 6 Network Diagram*

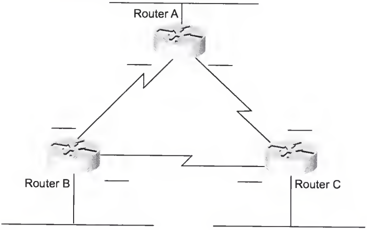


Table B-4 *Scenario 6 Planning Chart*

Location of Subnet Geographically	**Subnet Mask**	**Subnet Number**	**Router's IP Address**
Ethernet off Router A			
Ethernet off Router B			
Ethernet off Router C			
Serial between A and B			
Serial between A and C			
Serial between B and C			

Given the information in Figure B-7 and Table B-4, perform the following activities:

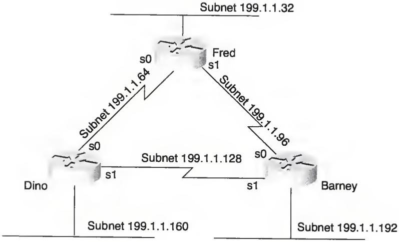
1. Determine all subnet masks that meet the criteria in the introduction to this scenario.
2. Choose a mask and pick enough subnets to use for the original topology (refer to Figure B-7).
3. Create IP-related configuration commands for each router.

Answers to Chapter 12 Scenarios

Answers to Scenario 5

Assuming that you had issued the commands in Example B-8, the most specific network diagram would look like Figure B-8.

Figure B-8 *Scenario 5 Answer—Network with Router Fred*



The clues that you should have found in the show commands are as follows:

- The **show interface** and **show ip interface brief** command output show the types of interfaces, as well as their IP addresses.
- The subnets could be learned from the **show ip route** command or derived from the IP addresses and masks shown in the **show interface** command output.
- The neighboring routers' IP addresses could be learned from the **show ip protocol** command.
- The neighboring routers' IP addresses and host names could be learned from the **show cdp neighbor detail** command.
- The metric for subnet 199.1.1.128/27 in RIP updates implies that both neighbors have an equal-cost route to the same subnet. Because two separate but duplicate networks would be a bad design, you can assume that the neighboring routers are attached to the same medium.

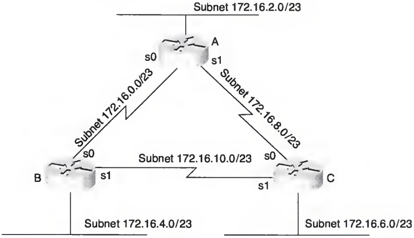
- If you are completely bored, the telnet **199.1.1.x** command could have been issued for all IP addresses in subnets not directly connected to Fred, hoping to get a router login prompt. That would identify the IP addresses of other router interfaces.

There is no way to know what physical media are beyond the neighboring routers. However, because CDP claims that both routers are 2500 series routers, the possible interfaces on these neighboring routers are limited. Figure B-8 shows the other subnets as Ethernet segments. Similarly, the figure shows the two neighboring routers attached to the same medium, which is shown as a serial link in Figure B-8.

Answers to Scenario 6

Figure B-9 shows one correct answer for the network skeleton presented in Figure B-7.

Figure B-9 *Scenario 6 Diagram Scratch Pad*



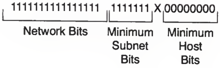
Answers to Task 1 for Scenario 6

Given the details in Figure B-7 and Table B-4 for Scenario 6, the subnet mask criteria are as follows:

- 200 hosts in a subnet, maximum
- 100 subnets, maximum
- Static size masks used all over this network

The mask must have at least eight host bits because 2^7 - 2 = 126 is not enough and 2^8 - 2 = 254 is more than enough for numbering 200 hosts in a subnet. The mask must have at least seven subnet bits, likewise, because 2^7 is the smallest power of 2 that is larger than 100, which is the required number of subnets. The first 16 bits in the mask must be binary 1 because a Class B network (172.16.0.0) is used. Figure B-10 diagrams the possibilities.

Figure B-10 *Subnet Mask Options for Scenario 6*



The only bit position in which a decision can be made is the 24th bit, shown with an X in Figure B-10. That leaves two mask possibilities: 255.255.254.0 and 255.255.255.0. This sample shows the 255.255.254.0 mask just so you can have a little more practice with harder masks. Given the choice in a real network, choose the easy mask!

Answers to Task 2 for Scenario 6

To choose a mask and pick enough subnets to use for the original topology illustrated in Figure B-7, a review of the longer binary algorithm and shortcut algorithm for deriving subnet numbers is required. To review, subnet numbers have the network number value in the network portion of the subnet numbers and have all binary 0s in the host bits. The bits that vary from subnet to subnet are the subnet bits—in other words, you are numbering different subnets in the subnet field.

Valid subnets with mask 255.255.254.0 are as follows:

```
172.16.0.0 (zero subnet)
172.16.2.0
172.16.4.0
172.16.6.0
.
.
.
172.16.252.0
172.16.254.0 (broadcast subnet)
```

The first six subnets, including the zero subnet, were chosen for this example, as listed in Table B-5.

Table B-5 *Scenario 6 Subnets and Addresses*

Location of Subnet Geographically	Subnet Mask	Subnet Number	Router's IP Address
Ethernet off Router A	255.255.254.0	172.16.2.0	172.16.2.1
Ethernet off Router B	255.255.254.0	172.16.4.0	172.16.4.2
Ethernet off Router C	255.255.254.0	172.16.6.0	172.16.6.3
Serial between A and B	255.255.254.0	172.16.0.0	172.16.0.1 (A) and .2 (B)
Serial between A and C	255.255.254.0	172.16.8.0	172.16.8.1 (A) and .3
Serial between B and C	255.255.254.0	172.16.10.0	172.16.10.2 (B) and .3

Answers to Task 3 for Scenario 6

Given the details in Figure B-7 and Table B-4 for Scenario 6, the configurations in Examples B-9 through B-11 satisfy the exercise of creating IP-related configuration commands for each router. These examples include only the IP-related commands.

Example B-9 *Router A Configuration, Scenario 6*

```
ip subnet-zero
no ip domain-lookup
!
interface serial0
ip address 172.16.0.1 255.255.254.0
interface serial 1
ip address 172.16.8.1 255.255.254.0
interface ethernet 0
ip address 172.16.2.1 255.255.254.0
!
router igrp 6
network 172.16.0.0
```

Example B-10 *Router B Configuration, Scenario 6*

```
ip subnet-zero
no ip domain-lookup
!
interface serial0
ip address 172.16.0.2 255.255.254.0
interface serial 1
ip address 172.16.10.2 255.255.254.0
interface ethernet 0
ip address 172.16.4.2 255.255.254.0
!
router igrp 6
network 172.16.0.0
```

Example B-11 *Router C Configuration, Scenario 6*

```
ip subnet-zero
no ip domain-lookup
!
interface serial0
ip address 172.16.8.3 255.255.254.0
interface serial 1
ip address 172.16.10.3  255.255.254.0
interface ethernet 0
ip address 172.16.6.3 255.255.254.0
!
router igrp 6
network 172.16.0.0
```

APPENDIX **C**

Hands-on Lab Exercises

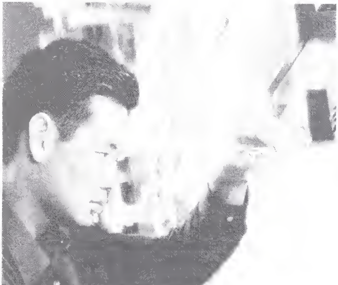
Some jobs require that you be able to configure routers and switches, but others do not. For instance, many people who sell Cisco products for Cisco or Cisco Channel Partners might have never configured a router or switch. However, those same people might know a lot more about other things, like the Cisco product line and what the latest products are. Simply put, some jobs require different skills and knowledge.

Cisco created CCNA as part of an overall plan to assess and verify the skill sets of the various Cisco Channel Partners. The CCNA certification's role was to prove the basic proficiency of a Channel Partner employee in network installation and support. Because CCNA focuses on network installation and support, Cisco wants CCNAs to be able to configure routers and switches.

In order to better test people on whether they have hands-on skills, Cisco includes a practical component on the CCNA exams. The exam engine will simulate routers and switches. Therefore, this chapter is designed to help you practice your hands-on skills.

Is it a good thing that Cisco is adding simulated labs to CCNA? Absolutely! If you have made it this far in the book, you definitely want to learn this stuff, not just pass a test. And the more Cisco can make the exams like a real implementation, asking you to apply the knowledge you have gained rather than just spew forth memorized answers, the more valuable the CCNA certification becomes. Wouldn't you rather configure a small IP network than memorize that pressing Esc-B backs up a single word in the command line? With more hands-on skills on the exam, there will be less time for trivial questions.

So how should you prepare? In a word, practice! Get some routers and switches and do the lab exercises in this chapter—even if you think you understand it all. You can use NetSim network simulator on the accompanying CD to do these labs. Also, use the exam engine on the CD to perform all the simulator-based practice questions. Build some "muscle memory" for implementing simple networks. Do you think Michael Jordan figured out how to shoot a jump shot when he was 12 and then quit shooting them? No, he practiced them a lot, so now his muscles remember what to do, and he doesn't think about how to shoot a jump shot every time he does it. So, the more you practice configuring routers and switches, the easier it will be to breeze through the exam instead of being hesitant about taking the exam with the new practical component.



Options for Gaining Hands-on Skills

When I sat down to write this chapter, I thought about the different options available when you want to develop hands-on skills. Each option has advantages and disadvantages:

- Borrowing gear from your company's test lab—If you already work in a networking job, chances are you can scrounge around and find some gear to use. Of course, getting a combination of gear that matches the examples in the books you're using might be a little more challenging, and collecting the gear each time you have a few hours to study might be a hassle. However, it's still one of the best ways to get hands-on skills. One key is to have some direction as to what to do with the gear after you get it. The labs in this appendix, and others as listed in Appendix C printed in the book, give you some example lab exercises to perform.
- Buying some gear on the Internet—You can always buy your own equipment if you do not have access to it at your job. It takes money, but you can always sell the gear when you're done, as long as it isn't broken or obsolete. Collecting a variety of gear to match a book's different examples and scenarios might be difficult, because most books' examples (this one included) are not written to minimize the amount of equipment needed to build the lab. The lab exercises in this chapter use the minimum amount of equipment needed to let you learn the necessary information, hopefully keeping the price down for you.
- Leasing the gear—Some companies will lease you a CCNA lab, but you still have the problem of collecting a variety of gear to match the different examples and scenarios in a book.
- Simulators—A special version of Boson's NetSim network simulator is included on the CD that comes with this book. This special version allows you to perform the labs in this appendix, as well as some of the other labs and scenarios—see Appendix C in the book for more details. Although simulators cannot teach everything, they are generally adequate for what you need to learn for CCNA.
- Lab rentals (e-labs)—Finally, several companies will rent lab time on lab pods accessible from the Internet. These typically can be rented by the hour or by the lab exercise, either for specific lab exercises or to do any labs you would like to perform. The labs in this book should work with some of the lab rental offerings.

About the Labs in This Appendix

There are three labs in this appendix. These three labs familiarize you with the CLI of routers and switches. They are designed for people who have not used routers and switches before, so they are very straightforward. They are also designed to be repeatable, until all the features and commands become second nature.

The CD that comes with the *CCNA ICND Exam Certification Guide*, provides three other labs on the CD. Those labs provide practice for the configuration and EXEC commands covered in throughout the book. Those labs assume that you know how to get around the user interface of the router and switch.

Equipment List

You will of course need some equipment in order to do the labs. For the labs in this chapter, you just need access to a single router (Lab 1), a single 2950 series switch (Lab 2), or two routers and one switch (Lab 3). Because different people might buy different equipment, this list describes the gear generically:

- **Routers (you need one for Labs 1 and 2, and a pair for Lab 3)**—Each router needs one Ethernet and one Serial interface. The serial interface can be a synchronous interface or an async/sync interface. Last time I checked these are priced around $200 to $300 on eBay.
- **2950 series switch (you need one)**—These switches are relatively new, being announced by Cisco in 2002. I found a few on Ebay for around $600 apiece, and you can generally get a brand new one for a little less than $1000.
- **Console kit (you need one)**—You need to access the console of the routers and switches. A console kit contains the correct cable and connector.
- **LAN cables (you need two)**—Two Category 5 straight-through cables attach the routers to the switch or hub. If your two routers use an AUI interface, you need AUI transceivers as well.
- **Serial cables**—You need one DCE cable and one DTE cable, which will be connected to create the serial link between R1 and R2. The connectors are dependent on the connectors on the routers you buy.

Using NetSim

You can use the NetSim network simulation software on the CD to perform the labs in this appendix. Appendix C in the book contains some hints and tips for loading NetSim, as well as listing some caveats when using NetSim to do labs from the book. Please refer to Appendix C in the (printed) book for more information.

List of Labs

Table C-1 describes the labs in this chapter. Please refer to Appendix C in the book for a list of all labs that can be done using the NetSim simulator.

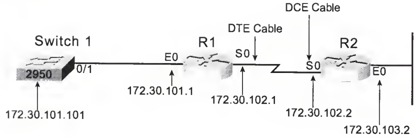
Table C-1 *Lab Descriptions*

Lab	Title	Description
1	Router Command-Line Interface Familiarization	The main goal of this lab is to make you familiar with getting around the router CLI. This lab offers you hints, ensuring that you understand the basics.
2	2950 Series Switch Command-Line Interface Familiarization	The main goal of this lab is to make you familiar with getting around the switch CLI. This lab offers you hints, ensuring that you understand the basics.
3	Basic Router IP Configuration and Management Navigation	This lab exercises your memory of basic IP configuration, as well as how to use the more popular IP troubleshooting commands.

Lab 1: Router Command-Line Interface Familiarization

All the labs in this chapter assume that the routers and switches have no existing configuration in them when the lab starts. Figure C-1 shows the network diagram used in most of the labs. In this lab, you only need to use R1.

Figure C-1 *Lab 1 Network*



Lab 1: Objectives

When finished with this lab, you will be able to do the following:

- Log in to a Cisco router via the console port
- Configure the passwords needed to log in via the console port and to enter privileged mode
- Get help via the router user interface
- Get help in EXEC and configuration modes

- Configure IP parameters
- Use several basic switch EXEC and configuration mode commands easily
- Navigate the different modes of a Cisco router CLI

This lab is intended to force you to try out several features of the CLI. You can, and should, branch out to try other commands. You should also repeat this lab until you've memorized all its commands and their syntax and you no longer need to ask for help to be able to remember the commands and their options. After the step-by-step instructions for this lab, you will get some hints, ensuring that you understand the basics.

NOTE You may want to get a scratch piece of paper on which to take notes at each step of the labs. Several steps ask you to record some specific information, so a piece of scratch paper will be helpful.

NOTE When using NetSim, after loading this lab from the NetSim lab navigator, you can start this lab at Step 6.

Lab 1: Step-by-Step Instructions

Step 1 Connect the console cable between your PC's COM1 port and a router's console port.

Step 2 Bring up your favorite terminal emulator program. If you do not have a favorite, use HyperTerminal, which comes with Microsoft operating systems. Select Start, Programs, Accessories, Communications, HyperTerminal.

Step 3 Ignore attempts to make you configure a phone number, but configure terminal characteristics of 9600 bps, 8 bits/byte, no parity, and 1 start/stop bit. This combination is often called 9600 8N1. It is what the router expects the console terminal to operate with.

Step 4 From the terminal emulator, press Enter.

Step 5 If you do not see a login prompt, repeat the preceding steps until you do.

Step 6 Using the help facilities described, look at the available commands.

Step 7 Try to enter privileged mode. Are you prompted for a password?

Step 8 When you are in privileged mode, use **help** to see the current list of commands.

Step 9 Guessing at some commands that look like they might be destructive or powerful, go back to user mode and use command help again to verify whether those commands are available in user mode.

Step 10 Repeat the last few steps until you have found three commands available in privileged mode but not user mode. List these commands.

Step 11 For the three commands recorded in step 10, use help to find the parameters for each command.

Step 12 How many serial interfaces are on this router? What are their names?

Step 13 How many packets and bytes have exited the lowest-numbered serial interface since the counters were last cleared? How long ago were the counters last cleared?

Step 14 What version of the Cisco IOS software is running?

Step 15 What is the name of the file in Flash memory?

Step 16 What was the time of and reason for the last reload?

Step 17 Enter configuration mode.

Using help, find the command that changes the router's host name.

Step 18 Change your host name to a name you like.

Step 19 Exit configuration mode.

Step 20 Using both old-style and new-style commands, verify that the RAM configuration and NVRAM configuration are different— namely, that the host name in NVRAM is the old name and the one in the RAM configuration is the new name you just specified.

Step 21 Enter configuration mode, and change the host name back to what it was previously.

Step 22 Save your configuration.

Step 23 Using help, find three global configuration commands that are interesting to you, and list them here.

Step 24 Enter interface configuration mode for the Ethernet interface.

Step 25 Using help, find two commands that are interesting to you, and list them here.

Step 26 In privileged EXEC mode, use the **clock set** command. Using help key sequences, type a syntactically correct **clock set** command. Then retrieve the command and move to the front of the command on the command line, back to the end, back one word, forward one word, back one letter, and forward one letter, each time using a single two-key combination.

Lab 1: Hints

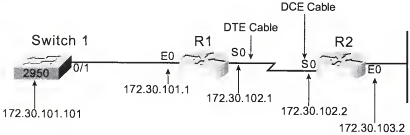
Table C-2 *Hints for Lab 1*

Step	Hint
1	I searched for "console connection" on www.cisco.com, and I found this pointer that might be helpful: www.cisco.com/univercd/cc/td/doc/product/access/acs_fix/cis2500/2520/2520_23/c2520ins.htm#xtocid894612.
4	If you do everything right but forget to press Enter, the router or switch will not write anything to the screen. You will not hurt anything by pressing Enter too many times!
5	Always try a different console cable and connector, make sure the cables are connected well, and make sure the cable is plugged into the console port, not the auxiliary port.
6	Use the **?** command.
7	Use the **enable** command.
8	Use the **?** command.
9	Use the **disable** and **enable** commands to move back and forth.
11	For example, if you choose **configure** as one of your commands, use the **configure ?** command to find information about the next option.
12	Use the **show interfaces** command.
13	Use the **show interfaces serial** *x* command.
14	Use the **show version** command.
15	Use the **show flash** command.
16	Use the **show version** command, and look closely!
17	Use the **configure terminal** command.
18	Use the **?** command to find the **hostname** command.
19	The right command would be something like **hostname Hannah**.
20	Press Ctrl-Z to exit, or repeatedly enter the **exit** command.
21	Use the **show startup-config** and **show running-config** commands (new) or the **show config** and **write terminal** commands (old).
23	Use the **copy running-config startup-config** command (new) or the **write memory** command (old).
24	Enter configuration mode first! Then, simply use the **?** command.
25	From global configuration mode, use the **interface ethernet** *x* command.
26	Some examples might be the **ip address** command and the **description** command.

Lab 2: 2950 Series Switch Command-Line Interface Familiarization

This lab assumes that the routers and switches have no existing configuration in them when the lab starts. Figure C-2 shows the network topology used in this lab.

Figure C-2 *Lab 2 Network*



Lab 2: Objectives

When finished with this lab, you will be able to do the following:

- Log in to a Cisco 2950 series switch via the console port
- Configure the passwords needed to log in via the console port and to enter privileged mode
- Get help in EXEC and configuration modes
- Configure IP parameters on the switch
- Use several basic switch EXEC and configuration mode commands easily
- Navigate the different modes of the Cisco 2950 series switch CLI

This lab, like Lab 1, is intended to force you to try out several features of the CLI. You can, and should, branch out to try other commands. You should also repeat this lab until you've memorized all its commands and their syntax and you no longer need to ask for help to be able to remember the commands and their options. After the step-by-step instructions for this lab, you will get some hints, ensuring that you understand the basics.

NOTE When using NetSim, after loading this lab from the NetSim lab navigator, you can start this lab at Step 7.

Lab 2: Step-by-Step Instructions

Step 1 Connect the console cable between your PC's COM1 port and a switch's console port.

Step 2 Bring up your favorite terminal emulator program. If you do not have a favorite, use HyperTerminal, which comes with Microsoft operating systems. Select Start, Programs, Accessories, Communications, HyperTerminal.

Step 3 Ignore attempts to make you configure a phone number, but configure terminal characteristics of 9600 bps, 8 bits/byte, no parity, and 1 start/stop bit. This combination is often called 9600 8N1. It is what the router expects the console terminal to operate with.

Step 4 Turn on the switch. The 2950 series has no on/off switch—you just plug it in.

Step 5 From the terminal emulator, press Enter.

Step 6 If you do not see a login prompt, repeat the preceding steps until you do.

Step 7 Using help facilities, look at the available commands. Do there seem to be more commands, or fewer commands, compared to the router CLI?

Step 8 Try to enter privileged mode. Are you prompted for a password?

Step 9 When you are in privileged mode, use **help** to see the current list of commands. Are there more commands than were shown in user mode? Are there more commands in switch CLI enable mode or router CLI enable mode?

Step 10 Guessing at some commands that look like they might be destructive or powerful, go back to user mode and look at command help again to verify whether those commands are unavailable in user mode.

Step 11 Repeat the last few steps until you have found three commands available in privileged mode but not user mode. List these commands.

Step 12 For the three commands you recorded in Step 11, use help to find the parameters for each command.

Step 13 How many LAN interfaces are on this switch? What are their names and/or numbers?

Step 14 How many packets and bytes have exited the lowest-numbered Ethernet interface since the counters were last cleared?

Step 15 What version of the switch IOS is running? Also note the "base MAC address" used by the switch (seen with the same command).

Step 16 What is the name of the file in Flash memory?

Step 17 What was the time of the last reload?

Step 18 Enter configuration mode.

Step 19 Using help, find the command that changes the switch's host name.

Step 20 Change your host name to a name that you like.

Step 21 Exit configuration mode.

Are the RAM configuration and NVRAM configuration different? List the commands you use to verify.

Step 22 Enter configuration mode, and change the host name back to what it was previously. If the host name was not set, set it to "switch1."

Step 23 Save your configuration. Record the command you use.

Step 24 Delete the saved configuration so that next time you boot the switch, the switch will use only default configuration parameters. What command did you use?

Step 25 Using help, find three global configuration commands that are interesting to you. List them here.

Step 26 Enter interface configuration mode for the Ethernet interface.

Step 27 Using help, find two commands that are interesting to you, and list them here.

Step 28 Set an enable password so that cisco must be typed when a user tries to enter privileged mode. What command did you use?

Step 29 Set the switch's IP address and default router based on Figure C-2. Record the commands you use.

Lab 2: Hints

Table C-3 *Hints for Lab 2*

Step	Hint
1	I searched for "console connection" on www.cisco.com, and found this pointer that may be helpful: www.cisco.com/univercd/cc/td/doc/product/access/acs_fix/cis2500/2520/2520_23/ c2520ins.htm#xtocid894612.
6	Always try a different console cable and connector, make sure the cables are connected well, and make sure the cables are plugged into the console port, not the aux port.
7	Use the ? command.
8	Use the **enable** command.
9	Use the ? command.

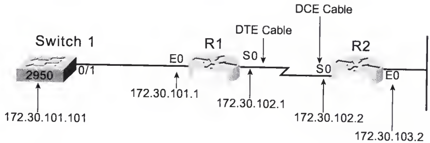
Table C-3 *Hints for Lab 2 (Continued)*

Step	Hint
10	Use the **disable** and **enable** commands to move back and forth.
13	Use the **show interfaces** command.
14	Use the **show interfaces fastethernet 0/1** command.
15, 17	Use the **show version** command.
18	Use the **configure terminal** command.
19	Use the **?** command to find the **hostname** command.
20	The right command would be something like **hostname Hannah**.
21	**Ctrl-z** exits, or repeated use of the **exit** command.
22	Use the **show running-config** command.
23	Use the **copy running-config startup-config** command.
24	Use the **erase startup-config** EXEC command.
26	Use the **interface ethernet 0/1** command.
28	Use the **enable password** or **enable secret** global config commands.

Lab 3: Basic Router IP Configuration and Management Navigation

This lab assumes that the routers and switches have no existing configuration in them when the lab starts. This lab also assumes that you know how to get around the user interface on the routers. Figure C-3 shows the network topology. If you are using two hubs, just cable the routers to the two hubs using straight-through Ethernet cables. If you are using a single switch, cable both routers to the switch, but first configure the switch to put the two ports in different VLANs.

Figure C-3 *Lab 3 Network*



Lab 3: Objectives

When finished with this lab, you will be able to do the following:

- Perform initial router configuration using setup mode
- Perform initial router configuration using configuration mode
- Verify IP connectivity using ping and trace
- Verify basic configuration using CDP
- Suspend and reconnect Telnet sessions

NOTE When using NetSim, after loading this lab from the NetSim lab navigator, you can start this lab at Step 3.

Lab 3: Step-by-Step Instructions

Step 1 Log in to R1, and use the write erase command to clear NVRAM.

Step 2 Issue the reload command to reload your router. What configuration will the router use when reloading?

Step 3 After the router has reloaded, you are asked if you want to enter the "initial configuration dialogue." Type yes.

Step 4 You are prompted with a series of commands. These commands expect a response from you. If there is a default answer, it is shown in brackets at the end of the command. You can just press Enter if that is the answer you want. Otherwise, type the appropriate answer, and the router creates the correct configuration for you.

Step 5 Configure a host name, configure all passwords as cisco, and configure IP parameters based on Figure C-3. All subnet masks are 255.255.255.0.

Step 6 When finished, you are asked if you want to use this configuration. If you are happy with what you configured, answer 2 and press Enter. If not, type 1, press Enter, and start again at Step 1. (When using NetSim, option 1 does not allow you to start over. Instead, choose this same lab from the NetSim lab navigator to re-start this lab.)

Step 7 Log in to R2, and use the write erase command to clear NVRAM.

Step 8 Issue the reload command to reload your router. What configuration will the router use when reloading?

Step 9 When the router has reloaded, you are asked if you want to enter the "initial configuration dialogue." Type no.

Step 10 Enter configuration mode, and configure R2. Give it a host name, configure all passwords as cisco, and configure IP addresses based on Figure C-3. All subnet masks are 255.255.255.0.

Step 11 On serial interface 0 (or your equivalent), configure the command clock rate 56000. This makes the router provide clocking, because there is no CSU/DSU in the sample network.

Step 12 Exit configuration mode, and save your configuration.

Step 13 Reconnect to R1's console. Using the **ping** command on R1, discover whether you can ping R2's IP addresses. List the IP addresses you can ping and those you cannot.

Step 14 Ping these same IP addresses using 1000-byte-long packets.

Step 15 Using the **trace** command on R1, discover the path taken to reach R2's IP addresses. (Hint: If **trace** never seems to want to finish, press **Ctrl-Shift-6** to stop the command.)

Step 16 Telnet to R2.

Step 17 Using the **ping** command on R2, discover whether you can ping R1's IP addresses. List the IP addresses you can ping and those you cannot.

Step 18 Ping these same IP addresses using 1000-byte-long packets.

Step 19 Using the **trace** command on R2, discover the path taken to reach R1's IP addresses. (Hint: If **trace** never seems to want to finish, press **Ctrl-Shift-6** to stop the command.)

Step 20 Suspend your Telnet connection. You should now be at the R1 command prompt.

Step 21 Look at the suspended Telnet connections on R1. What command did you use?

Step 22 Reconnect to R2 without having to type passwords again.

Step 23 On R2, exit (do not suspend) the Telnet connection.

Step 24 Back on R1, configure a host name for R2, referencing both IP addresses on R2.

Step 25 Telnet to R2 using the host name. Exit back to R1 when finished.

Step 26 On R1, use CDP to learn as much as you can about the neighboring switch and router without logging in to either device. What version of software is R2 using? Switch1? What are their IP addresses? What are their capabilities relative to each other?

Step 27 Save your configurations on both routers.

Lab Answers

Lab 1: Router Command Line Interface Familiarization

Example C-1 *Solution to Lab 1*

```
!
! Steps 1-3 are complete at this point, and the user pressed Enter
!
Router>?
Exec commands:
  access-enable    Create a temporary Access-List entry
  access-profile   Apply user-profile to interface
  clear            Reset functions
  connect          Open a terminal connection
  disable          Turn off privileged commands
  disconnect       Disconnect an existing network connection
  enable           Turn on privileged commands
  exit             Exit from the EXEC
  help             Description of the interactive help system
  lock             Lock the terminal
  login            Log in as a particular user
  logout           Exit from the EXEC
  mrinfo           Request neighbor and version information from a multicast
                   router
  mstat            Show statistics after multiple multicast traceroutes
  mtrace           Trace reverse multicast path from destination to source
  name-connection  Name an existing network connection
  pad              Open a X.29 PAD connection
  ping             Send echo messages
  ppp              Start IETF Point-to-Point Protocol (PPP)
  resume           Resume an active network connection
 --More
  rlogin           Open an rlogin connection
  show             Show running system information
  slip             Start Serial-line IP (SLIP)
  systat           Display information about terminal lines
  telnet           Open a telnet connection
  terminal         Set terminal line parameters
  traceroute       Trace route to destination
  tunnel           Open a tunnel connection
  udptn            Open an udptn connection
  where            List active connections
  x28              Become an X.28 PAD
  x3               Set X.3 parameters on PAD
```

Example C-1 *Solution to Lab 1 (Continued)*

```
!
! Next commands, steps 7 and 8
!
Router>enable
Router#?
Exec commands:
  access-enable     Create a temporary Access-List entry
  access-profile    Apply user-profile to interface
  access-template   Create a temporary Access-List entry
  alps              ALPS exec commands
  archive           manage archive files
  bfe               For manual emergency modes setting
  cd                Change current directory
  clear             Reset functions
  clock             Manage the system clock
  configure         Enter configuration mode
  connect           Open a terminal connection
  copy              Copy from one file to another
  debug             Debugging functions (see also 'undebug')
  delete            Delete a file
  dir               List files on a filesystem
  disable           Turn off privileged commands
  disconnect        Disconnect an existing network connection
  elog              Event-logging control commands
  enable            Turn on privileged commands
  erase             Erase a filesystem
  exit              Exit from the EXEC
 --More
  help              Description of the interactive help system
  isdn              Run an ISDN EXEC command on a BRI interface
  lock              Lock the terminal
  login             Log in as a particular user
  logout            Exit from the EXEC
  more              Display the contents of a file
  mrinfo            Request neighbor and version information from a multicast
                    router
  mrm               IP Multicast Routing Monitor Test
  mstat             Show statistics after multiple multicast traceroutes
  mtrace            Trace reverse multicast path from destination to source
  name-connection   Name an existing network connection
  ncia              Start/Stop NCIA Server
  no                Disable debugging functions
  pad               Open a X.29 PAD connection
  ping              Send echo messages
  ppp               Start IETF Point-to-Point Protocol (PPP)
  pwd               Display current working directory
  reload            Halt and perform a cold restart
```

continues

Example C-1 *Solution to Lab 1 (Continued)*

```
  restart           Restart Connection
  resume            Resume an active network connection
  rlogin            Open an rlogin connection
  rsh               Execute a remote command
 --More
  sdlc              Send SDLC test frames
  send              Send a message to other tty lines
  setup             Run the SETUP command facility
  show              Show running system information
  slip              Start Serial-line IP (SLIP)
  start-chat        Start a chat-script on a line
  systat            Display information about terminal lines
  telnet            Open a telnet connection
  terminal          Set terminal line parameters
  test              Test subsystems, memory, and interfaces
  traceroute        Trace route to destination
  tunnel            Open a tunnel connection
  udptn             Open an udptn connection
  undebug           Disable debugging functions (see also 'debug')
  verify            Verify a file
  where             List active connections
  write             Write running configuration to memory, network, or terminal
  x28               Become an X.28 PAD
  x3                Set X.3 parameters on PAD

!
! Next few lines are for step 10, with help (?) omitted
!
Router#disable
Router>enable
Router#disable
Router>enable
!
! I picked configure, reload, and show running-config
!
! Next command for step 12
!
Router#show interfaces
Ethernet0 is administratively down, line protocol is down
  Hardware is Lance, address is 0000.0c3e.5183 (bia 0000.0c3e.5183)
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 01:33:28, output 01:33:28, output hang never
  Last clearing of "show interface" counters never
```

Example C-1 *Solution to Lab 1 (Continued)*

```
 Queueing strategy: fifo
 Output queue 0/40, 0 drops; input queue 0/75, 0 drops
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 0 bits/sec, 0 packets/sec
    272 packets input, 140446 bytes, 0 no buffer
    Received 272 broadcasts, 0 runts, 0 giants, 0 throttles
    1 input errors, 0 CRC, 0 frame, 0 overrun, 1 ignored
    0 input packets with dribble condition detected
    417 packets output, 196326 bytes, 0 underruns(0/0/0)
    0 output errors, 0 collisions, 8 interface resets
    0 babbles, 0 late collision, 0 deferred
    0 lost carrier, 0 no carrier
    0 output buffer failures, 0 output buffers swapped out
--More
 Serial0 is administratively down, line protocol is down
 Hardware is HD64570
 MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
 Encapsulation HDLC, loopback not set
 Keepalive set (10 sec)
 Last input never, output 01:43:08, output hang never
 Last clearing of "show interface" counters 01:43:08
 Queueing strategy: fifo
 Output queue 0/40, 56 drops; input queue 0/75, 0 drops
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 0 bits/sec, 0 packets/sec
    0 packets input, 0 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    5 packets output, 850 bytes, 0 underruns
    0 output errors, 0 collisions, 2 interface resets
    0 output buffer failures, 0 output buffers swapped out
    0 carrier transitions
    DCD=up  DSR=up  DTR=down  RTS=down  CTS=up
--More
 Serial1 is administratively down, line protocol is down
 Hardware is HD64570
 MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
 Encapsulation HDLC, loopback not set
 Keepalive set (10 sec)
 Last input never, output 01:33:40, output hang never
 Last clearing of "show interface" counters never
 Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
 Queueing strategy: weighted fair
 Output queue: 0/1000/64/0 (size/max total/threshold/drops)
    Conversations  0/1/256 (active/max active/max total)
```

continues

Example C-1 *Solution to Lab 1 (Continued)*

```
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 1158 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     110 packets output, 2420 bytes, 0 underruns
     0 output errors, 0 collisions, 3 interface resets
     0 output buffer failures, 0 output buffers swapped out
     44 carrier transitions
 --More
  DCD=up  DSR=up  DTR=down  RTS=down  CTS=up
Router#show interface serial 0
Serial0 is administratively down, line protocol is down
  Hardware is HD64570
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
  Last input never, output 01:43:22, output hang never
  Last clearing of "show interface" counters 01:43:22
  Queueing strategy: fifo
  Dutput queue 0/40, 56 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     5 packets output, 850 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
     DCD=up  DSR=up  DTR=down  RTS=down  CTS=up
!
! Next command for step 14
!
Router#show version
Cisco Internetwork Operating System Software
IDS ™ 2500 Software (C2500-DS-L), Version 12.2(1), RELEASE SOFTWARE (fc2)
Copyright © 1986-2001 by cisco Systems, Inc.
Compiled Fri 27-Apr-01 14:43 by cmong
Image text-base: 0x03068DDC, data-base: 0x00001000

ROM: System Bootstrap, Version 5.2(5), RELEASE SOFTWARE
BDDTFLASH: 3000 Bootstrap Software (IGS-RXBOOT), Version 10.2(5), RELEASE SOFTWARE (fc1)
```

Example C-1 *Solution to Lab 1 (Continued)*

```
Router uptime is 1 hour, 46 minutes
System returned to ROM by reload
System image file is "flash:c2500-ds-l.122-1.bin"

cisco 2500 (68030) processor (revision D) with 16384K/2048K bytes of memory.
Processor board ID 01970904, with hardware revision 00000000
Bridging software.
X.25 software, Version 3.0.0.
1 Ethernet/IEEE 802.3 interface(s)
2 Serial network interface(s)
32K bytes of non-volatile configuration memory.
16384K bytes of processor board System flash (Read ONLY)

 --More
Configuration register is 0x2102

Router#show flash

System flash directory:
File  Length   Name/status
  1   13305352  c2500-ds-l.122-1.bin
[13305416 bytes used, 3471800 available, 16777216 total]
16384K bytes of processor board System flash (Read ONLY)

!
! Next command is only valid on some models of routers, to look at flash cards
!
Router#dir slot0:
           ^
% Invalid input detected at '^' marker.

Router#show version
Cisco Internetwork Operating System Software
IOS ™ 2500 Software (C2500-DS-L), Version 12.2(1), RELEASE SOFTWARE (fc2)
Copyright © 1986-2001 by cisco Systems, Inc.
Compiled Fri 27-Apr-01 14:43 by cmong
Image text-base: 0x03068DDC, data-base: 0x00001000

ROM: System Bootstrap, Version 5.2(5), RELEASE SOFTWARE
BOOTFLASH: 3000 Bootstrap Software (IGS-RXBOOT), Version 10.2(5), RELEASE SOFTWARE (fc1)

Router uptime is 1 hour, 46 minutes
System returned to ROM by reload
System image file is "flash:c2500-ds-l.122-1.bin"

cisco 2500 (68030) processor (revision D) with 16384K/2048K bytes of memory.
Processor board ID 01970904, with hardware revision 00000000
```

continues

Example C-1 *Solution to Lab 1 (Continued)*

```
Bridging software.
X.25 software, Version 3.0.0.
1 Ethernet/IEEE 802.3 interface(s)
2 Serial network interface(s)
32K bytes of non-volatile configuration memory.
16384K bytes of processor board System flash (Read ONLY)

 --More
Configuration register is 0x2102

!
! Next command for step 17
!
Router#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#?
Configure commands:
  aaa                         Authentication, Authorization and Accounting.
  access-list                 Add an access list entry
  alias                       Create command alias
  alps                        Configure Airline Protocol Support
  appletalk                   Appletalk global configuration commands
  arap                        Appletalk Remote Access Protocol
  arp                         Set a static ARP entry
  async-bootp                 Modify system bootp parameters
  autonomous-system           Specify local AS number to which we belong
  banner                      Define a login banner
  boot                        Modify system boot parameters
  bridge                      Bridge Group.
  bstun                       BSTUN global configuration commands
  buffers                     Adjust system buffer pool parameters
  busy-message                Display message when connection to host fails
  call-history-mib            Define call history mib parameters
  cdp                         Global CDP configuration subcommands
  chat-script                 Define a modem chat script
  class-map                   Configure QoS Class Map
  clock                       Configure time-of-day clock
  config-register             Define the configuration register
 --More
  connect                     cross-connect two interfaces
  decnet                      Global DECnet configuration subcommands
  default                     Set a command to its defaults
  default-value               Default character-bits values
  define                      interface range macro definition
  dialer                      Dialer commands
  dialer-list                 Create a dialer list entry
```

Example C-1 *Solution to Lab 1 (Continued)*

```
  dlsw                        Data Link Switching global configuration commands
  dnsix-dmdp                  Provide DMDP service for DNSIX
  dnsix-nat                   Provide DNSIX service for audit trails
  downward-compatible-config  Generate a configuration compatible with older
                              software
  dspu                        DownStream Physical Unit Command
  enable                      Modify enable password parameters
  end                         Exit from configure mode
  endnode                     SNA APPN endnode command
  exception                   Exception handling
  exit                        Exit from configure mode
  file                        Adjust file system parameters
  frame-relay                 global frame relay configuration commands
  help                        Description of the interactive help system
  hostname                    Set system's network name
  interface                   Select an interface to configure
--More
  ip                          Global IP configuration subcommands
  ipx                         Novell/IPX global configuration commands
  key                         Key management
  line                        Configure a terminal line
  lnm                         IBM Lan Manager
  locaddr-priority-list       Establish queueing priorities based on LU address
  location                    Network Management Router location Command
  logging                     Modify message logging facilities
  login-string                Define a host-specific login string
  map-class                   Configure static map class
  map-list                    Configure static map list
  menu                        Define a user-interface menu
  modemcap                    Modem Capabilities database
  mop                         Configure the DEC MOP Server
  multilink                   PPP multilink global configuration
  ncia                        Native Client Interface Architecture
  netbios                     NETBIOS access control filtering
  no                          Negate a command or set its defaults
  ntp                         Configure NTP
  parser                      Configure parser
  partition                   Partition device
  policy-map                  Configure QoS Policy Map
  printer                     Define an LPD printer
--More
  priority-list               Build a priority list
  privilege                   Command privilege parameters
  process-max-time            Maximum time for process to run before
                              voluntarily relinquishing processor
  prompt                      Set system's prompt
  queue-list                  Build a custom queue list
```

continues

Example C-1 *Solution to Lab 1 (Continued)*

```
  resume-string                Define a host-specific resume string
  rif                          Source-route RIF cache
  rlogin                       Rlogin configuration commands
  rmon                         Remote Monitoring
  route-map                    Create route-map or enter route-map command mode
  router                       Enable a routing process
  rsrb                         RSRB LSAP/DSAP filtering
  rtr                          RTR Base Configuration
  sap-priority-list            Establish queueing priorities based on SAP and/or
                               MAC address(es)
  scheduler                    Scheduler parameters
  service                      Modify use of network based services
  sgbp                         SGBP Stack Group Bidding Protocol configuration
  smrp                         Simple Multicast Routing Protocol configuration
                               commands
  sna                          Network Management Physical Unit Command
  snmp-server                  Modify SNMP parameters
 --More
  source-bridge                Source-route bridging ring groups
  standby                      Global HSRP configuration commands
  state-machine                Define a TCP dispatch state machine
  stun                         STUN global configuration commands
  subscriber-policy            Subscriber policy
  tacacs-server                Modify TACACS query parameters
  template                     Select a template to configure
  terminal-queue               Terminal queue commands
  tftp-server                  Provide TFTP service for netload requests
  time-range                   Define time range entries
  username                     Establish User Name Authentication
  virtual-profile              Virtual Profile configuration
  virtual-template             Virtual Template configuration
  vpdn                         Virtual Private Dialup Network
  vpdn-group                   VPDN group configuration
  x25                          X.25 Level 3
  x29                          X29 commands

Router(config)#hostname ?
  WORD  This system's network name

!
! Next commands for steps 19 and 20  notice the
! command prompt changes immediately
!
Router(config)#hostname Hannah
Hannah(config)#
Hannah(config)#exit
Hannah#
```

Example C-1 *Solution to Lab 1 (Continued)*

```
01:47:10: %SYS-5-CONFIG_I: Configured from console by console
Hannah#show running-config
Building configuration...

Current configuration : 531 bytes
!
version 12.2
no service single-slot-reload-enable
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Hannah
!
logging rate-limit console 10 except errors
!
ip subnet-zero
no ip finger
!
no ip dhcp-client network-discovery
!
!
!
interface Ethernet0
 no ip address
 shutdown
!
! Stopped the rest of the output of this command to save space. Step 21.
! Next command is at step 21
!
Hannah#show startup-config
%% Non-volatile configuration memory is not present
Hannah#write terminal
Building configuration...

Current configuration : 531 bytes
!
version 12.2
no service single-slot-reload-enable
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Hannah
!
logging rate-limit console 10 except errors
!
```

continues

Example C-1 *Solution to Lab 1 (Continued)*

```
ip subnet-zero
no ip finger
!
no ip dhcp-client network-discovery
!
!
!
interface Ethernet0
 no ip address
 shutdown
!
! Stopped the rest of the output of this command to save space. Step 21.
! Next command is at step 21
!
!
! The next command is still part of step 21.
! In most cases, you will see some old configuration when you use the show config
! command. In this case, no one has saved a configuration into NVRAM since
! it was cleared.
!
Hannah#show config
%% Non-volatile configuration memory is not present

!
! About to do steps 22 and 23.
!
Hannah#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
Hannah(config)#hostname R1
R1(config)#^Z
R1#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
R1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#?
Configure commands:
  aaa                         Authentication, Authorization and Accounting.
  access-list                 Add an access list entry
  alias                       Create command alias
  alps                        Configure Airline Protocol Support
  appletalk                   Appletalk global configuration commands
  arap                        Appletalk Remote Access Protocol
  arp                         Set a static ARP entry
  async-bootp                 Modify system bootp parameters
```

Example C-1 *Solution to Lab 1 (Continued)*

```
  autonomous-system           Specify local AS number to which we belong
  banner                      Define a login banner
  boot                        Modify system boot parameters
  bridge                      Bridge Group.
  bstun                       BSTUN global configuration commands
  buffers                     Adjust system buffer pool parameters
  busy-message                Display message when connection to host fails
  call-history-mib            Define call history mib parameters
  cdp                         Global CDP configuration subcommands
  chat-script                 Define a modem chat script
  class-map                   Configure QoS Class Map
  clock                       Configure time-of-day clock
  config-register             Define the configuration register
!
! Stopped output of help to save space. Step 24.
!
R1(config)#interface ethernet 0
R1(config-if)#?
Interface configuration commands:
  access-expression       Build a bridge boolean access expression
  appletalk               Appletalk interface subcommands
  arp                     Set arp type (arpa, probe, snap) or timeout
  backup                  Modify backup parameters
  bandwidth               Set bandwidth informational parameter
  bridge-group            Transparent bridging interface parameters
  carrier-delay           Specify delay for interface transitions
  cdp                     CDP interface subcommands
  cmns                    OSI CMNS
  custom-queue-list       Assign a custom queue list to an interface
  decnet                  Interface DECnet config commands
  default                 Set a command to its defaults
  delay                   Specify interface throughput delay
  description             Interface specific description
  dlsw                    DLSw interface subcommands
  dspu                    Down Stream PU
  exit                    Exit from interface configuration mode
  fair-queue              Enable Fair Queuing on an Interface
  fras                    DLC Switch Interface Command
  help                    Description of the interactive help system
  hold-
  ip                      Interface Internet Protocol config commands
  ipx                     Novell/IPX interface subcommands
  keepalive               Enable keepalive
  lan-name                LAN Name command
  llc2                    LLC2 Interface Subcommands
  load-interval           Specify interval for load calculation for an
                          interface
```

continues

Example C-1 *Solution to Lab 1 (Continued)*

```
  locaddr-priority        Assign a priority group
  logging                 Configure logging for interface
  loopback                Configure internal loopback on an interface
  mac-address             Manually set interface MAC address
  max-reserved-bandwidth  Maximum Reservable Bandwidth on an Interface
  media-type              Interface media type
  mop                     DEC MOP server commands
  mtu                     Set the interface Maximum Transmission Unit (MTU)
  multilink-group         Put interface in a multilink bundle
  netbios                 Use a defined NETBIOS access list or enable
                          name-caching

  no                      Negate a command or set its defaults
  ntp                     Configure NTP
  pppoe                   pppoe interface subcommands
  priority-group          Assign a priority group to an interface
  random-detect           Enable Weighted Random Early Detection (WRED) on an
 --More
  rate-limit              Rate Limit
  rmon                    Configure Remote Monitoring on an interface
  sap-priority            Assign a priority group
  service-policy          Configure QoS Service Policy
  shutdown                Shutdown the selected interface
  smrp                    Simple Multicast Routing Protocol interface
                          subcommands

  sna                     SNA pu configuration
  snapshot                Configure snapshot support on the interface
  snmp                    Modify SNMP interface parameters
  standby                 Interface HSRP configuration commands
  timeout                 Define timeout values for this interface
  traffic-shape           Enable Traffic Shaping on an Interface or
                          Sub-Interface

  transmit-interface      Assign a transmit interface to a receive-only
                          interface

  tx-ring-limit           Configure PA level transmit ring limit

R1(config-if)#exit
R1(config)#exit

!
! Starting step 27. Can't really show the command line manipulation.
!
R1#clock
01:49:19: %SYS-5-CONFIG_I: Configured from console by console
% Incomplete command.

R1#clock ?
  set  Set the time and date
```

Example C-1 *Solution to Lab 1 (Continued)*

```
R1#clock set ?
  hh:mm:ss  Current Time

R1#clock set 16:52:30 ?
  <1-31>  Day of the month
  MONTH   Month of the year

R1#clock set 16:52:30 31 dec ?
  <1993-2035>  Year

R1#clock set 16:52:30 31 dec 2001 ?
  <cr>

R1#clock set 16:52:30 31 dec 2001
```

Lab 2: 2950 Series Switch Command Line Interface Familiarization

Example C-2 *Solution to Lab 2*

```
!
! Screen capture begins at step 7, with the switch completing power-up
!
SW12>?
Exec commands:
  access-enable    Create a temporary Access-List entry
  clear            Reset functions
  connect          Open a terminal connection
  disable          Turn off privileged commands
  disconnect       Disconnect an existing network connection
  enable           Turn on privileged commands
  exit             Exit from the EXEC
  help             Description of the interactive help system
  lock             Lock the terminal
  login            Log in as a particular user
  logout           Exit from the EXEC
  mrinfo           Request neighbor and version information from a multicast router
  mstat            Show statistics after multiple multicast traceroutes
  mtrace           Trace reverse multicast path from destination to source
  name-connection  Name an existing network connection
  ping             Send echo messages
  rcommand         Run command on remote switch
```

continues

Example C-2 *Solution to Lab 2 (Continued)*

```
  resume            Resume an active network connection
  show              Show running system information
  systat            Display information about terminal lines
  telnet            Open a telnet connection
  terminal          Set terminal line parameters
  traceroute        Trace route to destination
  tunnel            Open a tunnel connection
  where             List active connections
SW12>enable


!
! Step 11 here, just didn't show repetitive "?" commands to save space.
!
SW12#?
Exec commands:
  access-enable     Create a temporary Access-List entry
  access-template   Create a temporary Access-List entry
  archive           manage archive files
  cd                Change current directory
  clear             Reset functions
  clock             Manage the system clock
  cns               CNS subsystem
  configure         Enter configuration mode
  connect           Open a terminal connection
  copy              Copy from one file to another
  debug             Debugging functions (see also 'undebug')
  delete            Delete a file
  dir               List files on a filesystem
  disable           Turn off privileged commands
  disconnect        Disconnect an existing network connection
  dot1x             IEEE 802.1X commands
  enable            Turn on privileged commands
  erase             Erase a filesystem
  exit              Exit from the EXEC
  format            Format a filesystem
  fsck              Fsck a filesystem
  help              Description of the interactive help system
  lock              Lock the terminal
  login             Log in as a particular user
  logout            Exit from the EXEC
  mkdir             Create new directory
  more              Display the contents of a file
  mrinfo            Request neighbor and version information from a multicast
                    router
  mrm               IP Multicast Routing Monitor Test
  mstat             Show statistics after multiple multicast traceroutes
  mtrace            Trace reverse multicast path from destination to source
```

Example C-2 *Solution to Lab 2 (Continued)*

```
  name-connection  Name an existing network connection
  no               Disable debugging functions
  ping             Send echo messages
  pwd              Display current working directory
  rcommand         Run command on remote switch
  reload           Halt and perform a cold restart
  rename           Rename a file
  resume           Resume an active network connection
  rmdir            Remove existing directory
  rsh              Execute a remote command
  send             Send a message to other tty lines
  setup            Run the SETUP command facility
  show             Show running system information
  systat           Display information about terminal lines
  telnet           Open a telnet connection
  terminal         Set terminal line parameters
  test             Test subsystems, memory, and interfaces
  traceroute       Trace route to destination
  tunnel           Open a tunnel connection
  udld             UDLD protocol commands
  undebug          Disable debugging functions (see also 'debug')
  verify           Verify a file
  vlan             Configure VLAN parameters
  vmps             VMPS actions
  vtp              Configure global VTP state
  where            List active connections
  write            Write running configuration to memory, network, or terminal

!
! Trying out a couple of examples for step 12 next.
!
SW12#copy ?
  bs:              Copy from bs: file system
  flash:           Copy from flash: file system
  ftp:             Copy from ftp: file system
  null:            Copy from null: file system
  nvram:           Copy from nvram: file system
  rcp:             Copy from rcp: file system
  running-config   Copy from current system configuration
  startup-config   Copy from startup configuration
  system:          Copy from system: file system
  tftp:            Copy from tftp: file system
  vb:              Copy from vb: file system
  xmodem:          Copy from xmodem: file system
  ymodem:          Copy from ymodem: file system
  zflash:          Copy from zflash: file system
```

continues

Example C-2 *Solution to Lab 2 (Continued)*

```
SW12#copy startup-config ?
  bs:             Copy to bs: file system
  flash:          Copy to flash: file system
  ftp:            Copy to ftp: file system
  null:           Copy to null: file system
  nvram:          Copy to nvram: file system
  rcp:            Copy to rcp: file system
  running-config  Update (merge with) current system configuration
  startup-config  Copy to startup configuration
  system:         Copy to system: file system
  tftp:           Copy to tftp: file system
  vb:             Copy to vb: file system
  xmodem:         Copy to xmodem: file system
  ymodem:         Copy to ymodem: file system
  zflash:         Copy to zflash: file system

SW12#copy startup-config running-config ?
  <cr>

SW12#copy startup-config running-config
Destination filename [running-config]?
1538 bytes copied in 0.312 secs
SW12#reload ?
  LINE    Reason for reload
  at      Reload at a specific time/date
  cancel  Cancel pending reload
  in      Reload after a time interval
  <cr>

SW12#reload in ?
Delay before reload (mmm or hhh:mm)

SW12#reload in 600 ?
  LINE  Reason for reload
  <cr>

SW12#reload in 600 cause I want to!

System configuration has been modified. Save? [yes/no]: y
Building configuration...
[OK]
Reload scheduled in 9 hours and 59 minutes
Reload reason: cause I want to!
Proceed with reload? [confirm]
SW12#
00:04:24: %SYS-5-SCHEDULED_RELOAD:  Reload requested for 10:04:07 UTC Mon Mar 1
1993 at  by console
```

Example C-2 *Solution to Lab 2 (Continued)*

```
SW12#erase ?
  flash:          Filesystem to be erased
  nvram:          Filesystem to be erased
  startup-config  Erase contents of configuration memory

SW12#erase nvram: ?
  <cr>

SW12#erase nvram:
Erasing the nvram filesystem will remove all files! Continue? [confirm]
[OK]
Erase of nvram: complete
!
! Step 13 next.
!

SW12#show interfaces status

Port    Name               Status       Vlan       Duplex  Speed Type
Fa0/1                      notconnect   1            auto    100 10/100BaseTX
Fa0/2                      connected    1          a-half   a-10 10/100BaseTX
Fa0/3                      connected    1          a-full  a-100 10/100BaseTX
Fa0/4                      connected    1          a-full  a-100 10/100BaseTX
Fa0/5                      connected    1          a-half   a-10 10/100BaseTX
Fa0/6                      notconnect   1            auto   auto 10/100BaseTX
Fa0/7                      notconnect   1            auto   auto 10/100BaseTX
Fa0/8                      notconnect   1            auto   auto 10/100BaseTX
Fa0/9                      notconnect   1            auto   auto 10/100BaseTX
Fa0/10                     notconnect   1            auto   auto 10/100BaseTX
Fa0/11                     notconnect   1            auto   auto 10/100BaseTX
Fa0/12                     notconnect   1            auto   auto 10/100BaseTX
Fa0/13                     notconnect   1            auto   auto 10/100BaseTX
Fa0/14                     notconnect   1            auto   auto 10/100BaseTX
Fa0/15                     notconnect   1            auto   auto 10/100BaseTX
Fa0/16                     notconnect   1            auto   auto 10/100BaseTX
Fa0/17                     notconnect   1            auto   auto 10/100BaseTX
Fa0/18                     notconnect   1            auto   auto 10/100BaseTX
Fa0/19                     notconnect   1            auto   auto 10/100BaseTX
Fa0/20                     notconnect   1            auto   auto 10/100BaseTX
Fa0/21                     notconnect   1            auto   auto 10/100BaseTX
Fa0/22                     notconnect   1            auto   auto 10/100BaseTX
Fa0/23                     notconnect   1            auto   auto 10/100BaseTX
Fa0/24                     notconnect   1            full    100 10/100BaseTX
Gi0/1                      notconnect   1            auto   auto unknown
Gi0/2                      notconnect   1            auto   auto unknown
!
! Step 15 next.
```

continues

Example C-2 *Solution to Lab 2 (Continued)*

```
!
SW12#show version
Cisco Internetwork Operating System Software
IOS (tm) C3550 Software (C3550-I5Q3L2-M), Version 12.1(11)EA1, RELEASE SOFTWARE
(fc1)
Copyright  1986-2002 by cisco Systems, Inc.
Compiled Wed 28-Aug-02 10:03 by antonino
Image text-base: 0x00003000, data-base: 0x0071D658

ROM: Bootstrap program is C3550 boot loader

SW12 uptime is 5 minutes
System returned to ROM by power-on
System image file is "flash:/c3550-i5q3l2-mz.121-11.EA1/c3550-i5q3l2-mz.121-11.EA1.bin"

cisco WS-C3550-24 (PowerPC) processor (revision E0) with 65526K/8192K bytes of memory.
Processor board ID CHK0635W02H
Last reset from warm-reset
Bridging software.
Running Layer2/3 Switching Image

Ethernet-controller 1 has 12 Fast Ethernet/IEEE 802.3 interfaces

Ethernet-controller 2 has 12 Fast Ethernet/IEEE 802.3 interfaces

Ethernet-controller 3 has 1 Gigabit Ethernet/IEEE 802.3 interface

Ethernet-controller 4 has 1 Gigabit Ethernet/IEEE 802.3 interface

24 FastEthernet/IEEE 802.3 interface(s)
2 Gigabit Ethernet/IEEE 802.3 interface(s)

The password-recovery mechanism is enabled.
384K bytes of flash-simulated non-volatile configuration memory.
Base ethernet MAC Address: 00:0A:B7:DC:B7:80
Motherboard assembly number: 73-5700-08
Power supply part number: 34-0966-02
Motherboard serial number: CAT0634058Q
Power supply serial number: DCA06340P5K
Model revision number: E0
Motherboard revision number: D0
Model number: WS-C3550-24-SMI
System serial number: CHK0635W02H
Configuration register is 0x10F

Reload scheduled in 9 hours and 58 minutes
Reload reason: cause I want to!
```

Example C-2 *Solution to Lab 2 (Continued)*

```
!
! Step 16 next.
!

SW12#show flash

Directory of flash:/

    2  -rwx           0   Mar 01 1993 00:10:52  env_vars
    3  -rwx           5   Mar 01 1993 00:04:13  private-config.text
    7  drwx         192   Mar 01 1993 01:23:36  c3550-i5q3l2-mz.121-11.EA1
   36  drwx          64   Mar 01 1993 00:00:31  crashinfo
    9  drwx         320   Mar 01 1993 01:00:38  c3550-i9q3l2-mz.121-9.EA1c
   38  -rwx         311   Mar 01 1993 00:10:52  system_env_vars

15998976 bytes total (6552064 bytes free)
!
! Step 18 next.
!

SW12#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SW12(config)#?
Configure commands:
  aaa                         Authentication, Authorization and Accounting.
  access-list                 Add an access list entry
  alias                       Create command alias
  arp                         Set a static ARP entry
  banner                      Define a login banner
  boot                        Boot Commands
  bridge                      Bridge Group.
  buffers                     Adjust system buffer pool parameters
  cdp                         Global CDP configuration subcommands
  class-map                   Configure QoS Class Map
  clock                       Configure time-of-day clock
  cluster                     Cluster configuration commands
  cns                         CNS Subsystem
  default                     Set a command to its defaults
  default-value               Default character-bits values
  define                      interface range macro definition
  dnsix-dmdp                  Provide DMDP service for DNSIX
  dnsix-nat                   Provide DNSIX service for audit trails
  do                          To run exec commands in config mode
  dot1x                       IEEE 802.1X subsystem
  downward-compatible-config  Generate a configuration compatible with older software
```

continues

Example C-2 *Solution to Lab 2 (Continued)*

```
  enable                      Modify enable password parameters
  end                         Exit from configure mode
  errdisable                  Error disable
  exception                   Exception handling
  exit                        Exit from configure mode
  file                        Adjust file system parameters
  help                        Description of the interactive help system
  hostname                    Set system's network name
  interface                   Select an interface to configure
  ip                          Global IP configuration subcommands
  key                         Key management
  l2protocol-tunnel           Tunnel Layer2 protocols
  line                        Configure a terminal line
  logging                     Modify message logging facilities
  mac                         Global MAC configuration subcommands
  mac-address-table           Configure the MAC address table
  map-class                   Configure static map class
  map-list                    Configure static map list
  mls                         Global Multi-Layer Switching parameters
  monitor                     Configure SPAN monitoring
  mvr                         Enable/Disable MVR on the switch
  no                          Negate a command or set its defaults
  ntp                         Configure NTP
  policy-map                  Configure QoS Policy Map
  port-channel                EtherChannel configuration
  priority-list               Build a priority list
  privilege                   Command privilege parameters
  process-max-time            Maximum time for process to run before
                              voluntarily relinquishing processor
  queue-list                  Build a custom queue list
  rmon                        Remote Monitoring
  route-map                   Create route-map or enter route-map command mode
  router                      Enable a routing process
  rtr                         RTR Base Configuration
  scheduler                   Scheduler parameters
  sdm                         Switch database management
  service                     Modify use of network based services
  shutdown                    Shutdown system elements
  snmp-server                 Modify SNMP parameters
  spanning-tree               Spanning Tree Subsystem
  subscriber-policy           Subscriber policy
  switchcore                  switchcore configuration
  system                      Set the system configuration
  tacacs-server               Modify TACACS query parameters
  template                    Select a template to configure
  tftp-server                 Provide TFTP service for netload requests
  time-range                  Define time range entries
```

Example C-2 *Solution to Lab 2 (Continued)*

```
  udld                          Configure global UDLD setting
  username                      Establish User Name Authentication
  vlan                          Vlan commands
  vmps                          VMPS settings
  vtp                           Configure global VTP state
!
! Step 20 next.
!

SW12(config)#hostname wendell
wendell(config)#exit
!
! Step 21 next.
!
wendell#show running-config
Building configuration...

Current configuration : 1568 bytes
!
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname wendell
!
!
ip subnet-zero
!
!
spanning-tree extend system-id
!
!
!
interface FastEthernet0/1
 no ip address
 speed 100
!
interface FastEthernet0/2
 no ip address
!
interface FastEthernet0/3
 no ip address
!
interface FastEthernet0/4
 no ip address
```

continues

Example C-2 *Solution to Lab 2 (Continued)*

```
!
interface FastEthernet0/5
 no ip address
!
interface FastEthernet0/6
 no ip address
!
interface FastEthernet0/7
 no ip address
!
interface FastEthernet0/8
 no ip address
!
interface FastEthernet0/9
 no ip address
!
interface FastEthernet0/10
 no ip address
!
interface FastEthernet0/11
 no ip address
!
interface FastEthernet0/12
 no ip address
!
interface FastEthernet0/13
 no ip address
!
interface FastEthernet0/14
 no ip address
!
interface FastEthernet0/15
 no ip address
!
interface FastEthernet0/16
 no ip address
!
interface FastEthernet0/17
 no ip address
!
interface FastEthernet0/18
 no ip address
!
interface FastEthernet0/19
 no ip address
!
interface FastEthernet0/20
```

Example C-2 *Solution to Lab 2 (Continued)*

```
 no ip address
!
interface FastEthernet0/21
 no ip address
!
interface FastEthernet0/22
 no ip address
!
interface FastEthernet0/23
 no ip address
!
interface FastEthernet0/24
 no ip address
 duplex full
 speed 100
!
interface GigabitEthernet0/1
 no ip address
!
interface GigabitEthernet0/2
 no ip address
!
interface Vlan1
 ip address 172.16.2.254 255.255.255.0
 shutdown
!
ip default-gateway 172.16.2.2
ip classless
ip http server
!
!
!
!
line con 0
line vty 0 4
 login
line vty 5 15
 login
!
end

wendell#show startup-config
%% Non-volatile configuration memory is not present

!
! Step 22 next.
!
```

continues

Example C-2 *Solution to Lab 2 (Continued)*

```
wendell#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
wendell(config)#hostname sw12
sw12(config)#^Z
!
! Step 23 next.
!
sw12#copy running-config startup-config
sw12#erase startup-config
!
! Step 25 next.
!
sw12(config)#?
Configure commands:
  aaa                          Authentication, Authorization and Accounting.
  access-list                  Add an access list entry
  alias                        Create command alias
  arp                          Set a static ARP entry
  banner                       Define a login banner
  boot                         Boot Commands
  bridge                       Bridge Group.
  buffers                      Adjust system buffer pool parameters
  cdp                          Global CDP configuration subcommands
  class-map                    Configure QoS Class Map
  clock                        Configure time-of-day clock
  cluster                      Cluster configuration commands
  cns                          CNS Subsystem
  default                      Set a command to its defaults
  default-value                Default character-bits values
  define                       interface range macro definition
  dnsix-dmdp                   Provide DMDP service for DNSIX
  dnsix-nat                    Provide DNSIX service for audit trails
  do                           To run exec commands in config mode
  dot1x                        IEEE 802.1X subsystem
  downward-compatible-config   Generate a configuration compatible with older
                               software
  enable                       Modify enable password parameters
  end                          Exit from configure mode
  errdisable                   Error disable
  exception                    Exception handling
  exit                         Exit from configure mode
  file                         Adjust file system parameters
  help                         Description of the interactive help system
  hostname                     Set system's network name
  interface                    Select an interface to configure
  ip                           Global IP configuration subcommands
  key                          Key management
```

Example C-2 *Solution to Lab 2 (Continued)*

```
  l2protocol-tunnel            Tunnel Layer2 protocols
  line                         Configure a terminal line
  logging                      Modify message logging facilities
  mac                          Global MAC configuration subcommands
  mac-address-table            Configure the MAC address table
  map-class                    Configure static map class
  map-list                     Configure static map list
  mls                          Global Multi-Layer Switching parameters
  monitor                      Configure SPAN monitoring
  mvr                          Enable/Disable MVR on the switch
  no                           Negate a command or set its defaults
  ntp                          Configure NTP
  policy-map                   Configure QoS Policy Map
  port-channel                 EtherChannel configuration
  priority-list                Build a priority list
  privilege                    Command privilege parameters
  process-max-time             Maximum time for process to run before
                               voluntarily relinquishing processor
  queue-list                   Build a custom queue list
  rmon                         Remote Monitoring
  route-map                    Create route-map or enter route-map command mode
  router                       Enable a routing process
  rtr                          RTR Base Configuration
  scheduler                    Scheduler parameters
  sdm                          Switch database management
  service                      Modify use of network based services
  shutdown                     Shutdown system elements
  snmp-server                  Modify SNMP parameters
  spanning-tree                Spanning Tree Subsystem
  subscriber-policy            Subscriber policy
  switchcore                   switchcore configuration
  system                       Set the system configuration
  tacacs-server                Modify TACACS query parameters
  template                     Select a template to configure
  tftp-server                  Provide TFTP service for netload requests
  time-range                   Define time range entries
  udld                         Configure global UDLD setting
  username                     Establish User Name Authentication
  vlan                         Vlan commands
  vmps                         VMPS settings
  vtp                          Configure global VTP state
!
! Step 26 next
!

sw12(config)#interface fastethernet 0/1
sw12(config-if)#?
```

continues

Example C-2 *Solution to Lab 2 (Continued)*

```
Interface configuration commands:
  arp                    Set arp type (arpa, probe, snap) or timeout
  bandwidth              Set bandwidth informational parameter
  bridge-group           Transparent bridging interface parameters
  carrier-delay          Specify delay for interface transitions
  cdp                    CDP interface subcommands
  channel-group          Etherchannel/port bundling configuration
  default                Set a command to its defaults
  delay                  Specify interface throughput delay
  description            Interface specific description
  dot1x                  IEEE 802.1X subsystem
  duplex                 Configure duplex operation.
  exit                   Exit from interface configuration mode
  flowcontrol            Configure flow operation.
  help                   Description of the interactive help system
  hold-queue             Set hold queue depth
  ip                     Interface Internet Protocol config commands
  keepalive              Enable keepalive
  l2protocol-tunnel      Tunnel Layer2 protocols
  load-interval          Specify interval for load calculation for an
                         interface
  logging                Configure logging for interface
  mac                    MAC interface commands
  mac-address            Manually set interface MAC address
  max-reserved-bandwidth Maximum Reservable Bandwidth on an Interface
  mls                    Configure MultiLayer Switching characteristics
  mvr                    MVR per port configuration
  no                     Negate a command or set its defaults
  ntp                    Configure NTP
  pagp                   PAgP interface subcommands
  priority-queue         Configure priority scheduling
  random-detect          Enable Weighted Random Early Detection (WRED) on an Interface
  rmon                   Configure Remote Monitoring on an interface
  service-policy         Configure QoS Service Policy
  shutdown               Shutdown the selected interface
  snmp                   Modify SNMP interface parameters
  spanning-tree          Spanning Tree Subsystem
  speed                  Configure speed operation.
  storm-control          storm configuration
  switchport             Set switching mode characteristics
  tcam                   tcam keyword
  timeout                Define timeout values for this interface
  transmit-interface     Assign a transmit interface to a receive-only
                         interface
  tx-ring-limit          Configure PA level transmit ring limit
  udld                   Configure UDLD enabled or disabled and ignore global
                         UDLD setting
```

Example C-2 *Solution to Lab 2 (Continued)*

```
  wrr-queue               Configure weighted round-robin xmt queues
!
! Step 27 next
!
sw12(config-if)#exit
sw12(config)#enable secret cisco
sw12(config)#interface vlan 1
sw12(config-if)#ip address 172.30.101.101 255.255.255.0
sw12(config-if)#exit
sw12(config)#ip default-gateway 172.30.101.1
sw12(config)#^Z
```

Lab 3: Basic Router IP Configuration and Management Navigation

Example C-3 *Solution to Lab 3*

```
R1#write erase
Erasing the nvram filesystem will remove all files! Continue? [confirm]
[OK]
Erase of nvram: complete
R1#reload
Proceed with reload? [confirm]

00:41:35: %SYS-5-RELOAD: Reload requested
System Bootstrap, Version 5.2(5), RELEASE SOFTWARE
Copyright © 1986-1994 by cisco Systems
2500 processor with 16384 Kbytes of main memory

ERR: Invalid chip id 0x80B5 (reversed = 0x1AD ) detected in System flash
F3: 12640836+664484+868488 at 0x3000060

              Restricted Rights Legend

Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
© of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
© (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

           cisco Systems, Inc.
           170 West Tasman Drive
           San Jose, California 95134-1706
```

continues

Example C-3 *Solution to Lab 3 (Continued)*

```
Cisco Internetwork Operating System Software
IOS ™ 2500 Software (C2500-DS-L), Version 12.2(1), RELEASE SOFTWARE (fc2)
Copyright © 1986-2001 by cisco Systems, Inc.
Compiled Fri 27-Apr-01 14:43 by cmong
Image text-base: 0x03068DDC, data-base: 0x00001000

cisco 2500 (68030) processor (revision D) with 16384K/2048K bytes of memory.
Processor board ID 01970904, with hardware revision 00000000
Bridging software.
X.25 software, Version 3.0.0.
1 Ethernet/IEEE 802.3 interface(s)
2 Serial network interface(s)
32K bytes of non-volatile configuration memory.
16384K bytes of processor board System flash (Read ONLY)

--- System Configuration Dialog ---

!
! About to start setup mode in R1, step 3.
!
Would you like to enter the initial configuration dialog? [yes/no]: yes
At any point you may enter a question mark '?' for help.
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.Basic management setup configures only enough
connectivity
for management of the system, extended setup will ask you
to configure each interface on the system

Would you like to enter basic management setup? [yes/no]: no
First, would you like to see the current interface summary? [yes]:
Any interface listed with OK? value "NO" does not have a valid configuration

Interface                  IP-Address      OK? Method Status                Protocol
Ethernet0                  unassigned      NO  unset  up                    down
Serial0                    unassigned      NO  unset  down                  down
Serial1                    unassigned      NO  unset  down                  down

Configuring global parameters:

  Enter host name [Router]: R1
The enable secret is a password used to protect access to
  privileged EXEC and configuration modes. This password, after
  entered, becomes encrypted in the configuration.
  Enter enable secret: cisco
The enable password is used when you do not specify an
  enable secret password, with some older software versions, and
  some boot images.
```

Example C-3 *Solution to Lab 3 (Continued)*

```
  Enter enable password: fred
The virtual terminal password is used to protect
  access to the router over a network interface.
  Enter virtual terminal password: cisco
Configure SNMP Network Management? [yes]: no
Configure bridging? [no]:
Configure DECnet? [no]:
Configure AppleTalk? [no]:
Configure IPX? [no]:
Configure IP? [yes]:
Configure IGRP routing? [yes]: no
Configuring interface parameters:
Do you want to configure Ethernet0  interface? [yes]:
Configure IP on this interface? [yes]:
IP address for this interface: 172.30.101.1
Subnet mask for this interface [255.255.0.0] : 255.255.255.0
Class B network is 172.16.0.0, 24 subnet bits; mask is /24
Do you want to configure Serial0  interface? [yes]:
  Configure IP on this interface? [yes]:
Configure IP unnumbered on this interface? [no]:
IP address for this interface: 172.30.102.1
Subnet mask for this interface [255.255.0.0] : 255.255.255.0
Class B network is 172.16.0.0, 24 subnet bits; mask is /24
Do you want to configure Serial1  interface? [yes]: no

The following configuration command script was created:

hostname R1
enable secret 5 $1$VOLh$pkIe0Xjx2sgjgZ/Y6Gt1s.
enable password fred
line vty 0 4
password cisco
no snmp-server
!
no bridge 1
no decnet routing
no appletalk routing
no ipx routing
ip routing
!
interface Ethernet0
ip address 172.30.101.1 255.255.255.0
no mop enabled
!
interface Serial0
ip address 172.30.102.1 255.255.255.0
no mop enabled
!
```

continues

Example C-3 *Solution to Lab 3 (Continued)*

```
interface Serial1
no ip address
no mop enabled
!
end

[0] Go to the IOS command prompt without saving this config.
[1] Return back to the setup without saving this config.
[2] Save this configuration to nvram and exit.

Enter your selection [2]: 2
Building configuration...
[OK]Use the enabled mode 'configure' command to modify this configuration.
Press RETURN to get started!

!
! Finished with step 6, now moving console cable to R2
!
R2#write erase
Erasing the nvram filesystem will remove all files! Continue? [confirm]
[OK]
Erase of nvram: complete
R2#reload
Proceed with reload? [confirm]

00:41:35: %SYS-5-RELOAD: Reload requested
System Bootstrap, Version 5.2(5), RELEASE SOFTWARE
!
!Omitted boot messages to save space.
!
--- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: no

Press Return to get started!

!
!About to start step 10.
!
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R2
R2(config)#enable secret cisco
R2(config)#enable password fred
R2(config)#line vty 0 4
R2(config-line)#login
R2(config-line)#password cisco
```

Example C-3 *Solution to Lab 3 (Continued)*

```
R2(config-line)#interface ethernet 0
R2(config-if)#ip address 172.30.103.2 255.255.255.0
R2(config-if)#interface serial 0
R2(config-if)#ip address 172.30.102.2 255.255.255.0
R2(config-if)#clock rate 56000
R2(config-if)#no shut
R2(config-if)#
01:25:24: %LINK-3-UPDOWN: Interface Serial0, changed state to up
01:25:25: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed state to up
R2(config-if)#^Z
R2#
!
!Just finished step 11, now to step 12. Still in R2's console.
!
R2#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
R2#
!
! Moving console cable back to R1. Step 13 next.
!
R1#
R1#show interface serial 0
Serial0 is up, line protocol is up
  Hardware is HD64570
  Internet address is 172.30.102.1/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
  Last input 00:00:06, output 00:00:07, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/1/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 1158 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     12 packets input, 1312 bytes, 0 no buffer
     Received 12 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     10 packets output, 726 bytes, 0 underruns
     0 output errors, 0 collisions, 6 interface resets
     0 output buffer failures, 0 output buffers swapped out
 --More
```

continues

Example C-3 *Solution to Lab 3 (Continued)*

```
     1 carrier transitions
     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
R1#ping 172.30.102.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.30.102.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/35/36 ms
R1#ping 172.30.103.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.30.103.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
!
! Ping from R1 to R2's Ethernet IP address does not work, because there
! are no routes in R1's routing table to the 172.30.103.0 subnet.
! Static routes, and then RIP and IGRP, will be configured in the
! labs on the CD in the CCNA ICND Exam Certification Guide.
!
R1#ping
Protocol [ip]:
Target IP address: 172.30.102.2
Repeat count [5]:
Datagram size [100]: 1000
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 1000-byte ICMP Echos to 172.30.102.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 292/292/292 ms
R1#ping
Protocol [ip]:
Target IP address: 172.30.103.2
Repeat count [5]:
Datagram size [100]: 1000
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 1000-byte ICMP Echos to 172.30.103.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
R1#trace 172.30.102.2

Type escape sequence to abort.
Tracing the route to 172.30.102.2
```

Example C-3 *Solution to Lab 3 (Continued)*

```
  1 172.30.102.2 28 msec 16 msec *
R1#trace 172.30.103.2

Type escape sequence to abort.
Tracing the route to 172.30.103.2

  1  *  *  *
  2  *  *  *
  3  *  *  *
  4
!
! Ctrl-Shift-6, x should break you out of the trace command when
! it wants to loop forever. Step 16 next.
!
R1#telnet 172.30.102.2
Trying 172.30.102.2 ... Open


User Access Verification

Password:
R2>enable
Password:
R2#ping 172.30.102.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.30.102.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 72/80/104 ms
R2#ping 172.30.101.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.30.101.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
!
! Ping from R2 to R1's Ethernet IP address does not work, because there
! are no routes in R2's routing table to the 172.30.101.0 subnet.
! Static routes, and then RIP and IGRP, will be configured in the
! next lab.
!
R2#ping
Protocol [ip]:
Target IP address: 172.30.102.1
Repeat count [5]:
Datagram size [100]: 1000
Timeout in seconds [2]:
```

continues

Example C-3 *Solution to Lab 3 (Continued)*

```
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 1000-byte ICMP Echos to 172.30.102.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 296/301/316 ms
R2#ping
Protocol [ip]:
Target IP address: 172.30.101.1
Repeat count [5]:
Datagram size [100]: 1000
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 1000-byte ICMP Echos to 172.30.101.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
R2#trace 172.30.102.1

Type escape sequence to abort.
Tracing the route to 172.30.102.1

  1 172.30.102.1 24 msec 28 msec *
R2#trace 172.30.101.1

Type escape sequence to abort.
Tracing the route to 172.30.101.1

  1  *  *  *
  2  *  *  *
  3  *  *
!
!Step 20 next, with a few extra related commands.
!
R2#show users
    Line       User       Host(s)              Idle       Location
   0 con 0                idle                 00:05:49
*  2 vty 0                idle                 00:00:00
R2#
R2#
!
! Pressed Ctrl-Shift-6, x next, and suspended back to R1.
!
R1#show users
    Line       User       Host(s)              Idle       Location
*  0 con 0                172.30.102.2         00:00:06
```

Example C-3 *Solution to Lab 3 (Continued)*

```
  Interface      User        Mode                     Idle     Peer Address

R1#show sessions
Conn Host                Address             Byte  Idle Conn Name
*  1 172.30.102.2        172.30.102.2           0     0 172.30.102.2

!
! the 1 command means to reconnect to the session number 1.
! Step 22.
!
R1#1
[Resuming connection 1 to 172.30.102.2 ... ]

R2#
R2#exit

[Connection to 172.30.102.2 closed by foreign host]
R1#show sessions
% No connections open
!
! Now on to step 24.
!
R1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#ip host ?
  WORD  Name of host

R1(config)#ip host R2 ?
  <0-65535>   Default telnet port number
  A.B.C.D     Host IP address
  additional  Append addresses

R1(config)#ip host R2 172.30.102.2 ?
  A.B.C.D  Host IP address
  <cr>

R1(config)#ip host R2 172.30.102.2 172.30.103.2
R1(config)#^Z
R1#telnet r2
Trying R2 (172.30.102.2)... Open


User Access Verification

Password:
R2>enable
```

continues

Example C-3 *Solution to Lab 3 (Continued)*

```
Password:
R2#exit

[Connection to r2 closed by foreign host]

!
! Step 26 next.
!
R1#show cdp ?
  entry      Information for specific neighbor entry
  interface  CDP interface status and configuration
  neighbors  CDP neighbor entries
  traffic    CDP statistics
  |          Output modifiers
  <cr>

R1#show cdp entry ?
  *     all CDP neighbor entries
  WORD  Name of CDP neighbor entry

R1#show cdp entry *
-------------------------
Device ID: 001029DE7BC0
Entry address(es):
  IP address: 172.30.101.101
Platform: cisco 1900,  Capabilities: Trans-Bridge Switch
Interface: Ethernet0,  Port ID (outgoing port): 1
Holdtime : 177 sec

Version :
V9.00

advertisement version: 2
Protocol Hello:  OUI=0x00000C, Protocol ID=0x0112; payload len=25,
value=00000000FFFFFFFF0101050000000000000000001029DE7BC0FF
VTP Management Domain: ''
Duplex: half

-------------------------
Device ID: R2
Entry address(es):
  IP address: 172.30.102.2
Platform: cisco 2500,  Capabilities: Router
Interface: Serial0,  Port ID (outgoing port): Serial0
 --More
Holdtime : 159 sec

Version :
```

Example C-3 *Solution to Lab 3 (Continued)*

```
Cisco Internetwork Operating System Software
IOS ™ 2500 Software (C2500-DS-L), Version 12.2(3), RELEASE SOFTWARE (fc1)
Copyright © 1986-2001 by cisco Systems, Inc.
Compiled Wed 18-Jul-01 21:10 by pwade

advertisement version: 2

R1#show cdp neighbor
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
001029DE7BC0     Eth 0              170         T S       1900      1
R2               Ser 0              152          R        2500      Ser 0
R1#show cdp interface
Ethernet0 is up, line protocol is up
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
Serial0 is up, line protocol is up
  Encapsulation HDLC
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
Serial1 is administratively down, line protocol is down
  Encapsulation HDLC
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
!
! Final step, step 27, saving configurations.
!
R1#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
R1#r2
Trying R2 (172.30.102.2)... Open


User Access Verification

Password:
R2>ena
Password:
R2#write memory
Building configuration...
[OK]
R2#
```

APPENDIX G

Which Routing Protocol?

Among all the thorny questions that network engineers are asked on a regular basis, probably among the hardest is this one:

> My nctwork currently runs Enhanced Interior Gateway Routing Protocol (EIGRP). Would I be better off if I switched to Open Shortest Path First (OSPF)?

You can replace the two protocols mentioned in this sentence with any pair of protocols among the advanced interior gateway protocols (OSPF, Intermediate System-to-Intermcdiatc System [IS-IS] and EIGRP), and you have described a question that routing protocol engineers are asked probably thousands of times a year. Of course, convergence is always faster on the other side of thc autonomous system boundary, so to speak, so it is always tempting to jump to another protocol as soon as a problem crops up with the one you arc running.

How do you answer this question in real life? You could try the standard, "It depends," but does this really answer the qucstion? The tactic in the Routing Protocols Escalation Team was to ask them questions until they went away, but none of these answers really helps the network operator or designer really answer thc question, "How do you decide which protocol is the best?"

Three questions are embedded within this question, really, and it is easier to think about them independently:

- Is one protocol, in absolute terms, "better" than all the other protocols, in all situations?
- If the answer to this first question is "No," does each routing protocol exhibit some set of characteristics that indicate it would fit some situations (specifically, network topologies) better than others?
- After you have laid out the basics, what is the tradeoff in living with what you currently have versus switching to another routing protocol? What factors do you need to consider when doing the cost/benefit analysis involved in switching from one routing protocol to another?

This appcndix takes you through each of thcsc three questions. This might be thc first and last time that you hear a network engineer actually answer the question, "Which routing protocol should I use?" so get rcady for a whirlwind tour through the world of routing.

[redacted]

Is One Protocol "Better" Than the Others?

The first thing you need to do with this sort of question is to qualify it: "What do you mean by better?" Some protocols are easier to configure and manage, others are easier to troubleshoot, some are more flcxible, and so on. Which one arc you going to look at?

This appendix examines ease of troubleshooting and convergence time. You could choose any number of other measures, including these:

- **Ease of management**—What do the Management Information Bases (MIBs) of the protocol cover? What sorts of **show** commands are available for taking a network baseline?
- **Ease of eonfiguration**—How many commands will the average configuration require in your network configuration? Is it possible to configure several routers in your network with the same configuration?
- **On-the-wire effieieney**—How much bandwidth does the routing protocol take up while in steady state, and how much could it take up, at most, when converging in response to a major network event?

Ease of Troubleshooting

The average uptime (or reliability) of a network is affected by two elements:

- How often does thc nctwork fail?
- How long does it take to recover from a failure?

The network design and your choice of equipment (not just the vendor and operating system, but also putting the right piecc of cquipment into each role and making certain that each device has enough memory, and so on) play heavily into the first element. The design of the network also plays into the second element. The piece often forgotten about when considering the reliability of a network is how long it takes to find and fix, or troubleshoot, the network when it fails.

Ease of management plays a role in the ease of troubleshooting, of course; if it is hard to take a baselinc of what the nctwork is supposed to look likc, you will not do so on a regular basis, and you will have a dated picture to troubleshoot from. The tools available for troubleshooting are also important. Of course, this is going to vary between the implementations of thc protocols; here, implementations in Cisco IOS Software illustrate the concepts. Table G-1 outlines some of the troubleshooting tools that are available in EIGRP, OSPF, and IS-IS, in Cisco IOS Software.

Table G-1 *Cisco IOS Software Troubleshooting Tools for EIGRP, OSPF, and IS-IS*

	EIGRP	OSPF	IS-IS
Debug Neighbors	Neighbor formation state; hello packets.	Neighbor formation state; hello packets.	Packets exchanged during neighbor formation.
Log Neighbor State	Yes.	Yes.	No.
Debug Database Exchange and Packets	Packets exchanged (updates, replies, and so on), with filters per neighbor or for a specific route.	Packets flooded, with filters for specific routing information. Packets retransmitted.	Packets flooded.
Debug Interactions with the Routing Table	Yes.	No.	No.
Debug Route Selection Process	Yes (DUAL[1] FSM[2] events).	Yes (SPF[3] events).	Yes (SPF events).
Show Database	Yes, by specific route and route state.	Yes, by LSA[4] type and advertising router.	Yes, by LSP[5] ID or type of route.
Event Log	Yes; understandable if you comprehend DUAL and its associated terminology.	Yes; only understandable if you have access to the source code.	No.

[1] DUAL = Diffusing Update Algorithm

[2] FSM = finite state machine

[3] SPF = shortest path first

[4] LSA = link-state advertisement

[5] LSP = link-state packet

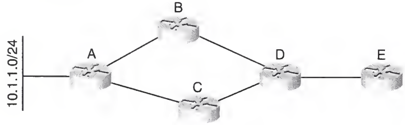
From this chart, you can see that EIGRP generally provides the most tools for finding a problem in the network quickly, with OSPF running a close second.

Which Protocol Converges Faster?

I was once challenged with the statement, "There is no way that a distance vector protocol can ever converge faster than a link-state protocol!" An hour and a half later, I think the conversation tapered off into, "Well, in some situations, I *suppose* a distance vector protocol *could* converge as fast as a link-state protocol," said without a lot of conviction.

In fact, just about every network engineer can point to reasons why he thinks a specific routing protocol will *always* converge faster than some other protocol, but the reality is that all routing protocols can converge quickly or slowly, depending on a lot of factors strictly related to network design, without even considering the hardware, types of links, and other random factors that play into convergence speed in different ways with each protocol. As a specific example, look at the small nctwork illustrated in Figure G-1 and consider the various options and factors that might play into convergence speed in this network.

Figure G-1 *Simple Network*



This figure purposefully has no labels showing anything concerning routing protocols configuration or design; instead, this section covers several possible routing configurations and examines how the same protocol could converge more or less quickly even on a network this small through just minor configuration changes.

Start with EIGRP as an example:

- The Router A to C link has a bandwidth of 64 kbps.
- The Router A to B link has a bandwidth of 10 Mbps.
- The Router B to D and Router C to D links have equal bandwidths.

With this information in hand, you can determine that Router D is going to mark the path to 10.1.1.0/24 through Router B as the best path (the *successor* in EIGRP terms). The path through Router C will not be marked as a *feasible successor*, because the differential in the metrics is too great between the two paths. To the EIGRP process running on Router D, the path through Router C cannot be proven based on the metrics advertised by Routers B and C, so the path through Router C will not be installed as a possible backup route.

This means that if the Router B to D link fails, Router D is forced to mark 10.1.1.0/24 as *active* and scnd a query to Router C. The convergence time is bounded by the amount of time it takes for the following tasks:

- Routcr D to examine its local topology table and determine that no other known loop-frce paths exist.
- Router D to build and transmit a query toward Router C.
- Router C to receive and process the query, including examining its local EIGRP topology table, and find it still has an alternate path.
- Router C to build a reply to the query and transmit it.
- Router D to receive the reply and process it, including route installation time and the time required to change the information in the forwarding tables on the router.

Many factors are contained in these steps; any one of them could take a long time. In the real world, the total time to complete the steps in this network is less than two or three seconds.

Now change the assumptions just slightly and see what the impact is:

- The Router A to C link and A to B links have equal bandwidth.
- The Router B to D link has a bandwidth of 64 kbps.
- The Router B to C link has a bandwidth of 10 Mbps.

As you can tell, the network conditions have been changed only slightly, but the results are altered dramatically. In this case, the path to 10.1.1.0/24 through Router C is chosen as the best path. EIGRP then examines the path through Router B and finds that it is a loop-free path, based on the information embedded in EIGRP metrics. What happens if the Router B to C link fails?

The process has exactly one step: Router D examines its local EIGRP topology table and finds that an alternate loop-free path is available. Router D installs this alternate route in the local routing table and alters the forwarding information as needed. This processing takes on the order of 150 milliseconds or less.

Using the same network, examine the various reactions of OSPF to link failures. Begin with these:

- The Router B to D link has a cost of 20.
- All other links in the network have a cost of 10.
- All routes are internal OSPF routes.

What happens if the Router B to C link fails?

1. Router B and C detect the link failure and wait some period of time, called the link-state advertisement (LSA) generation time. Then they flood modified router LSAs with this information.
2. The remaining routers in the network receive this new LSA and place it in their local link-state databases. The routers wait some period of time, called the shortest path first (SPF) wait time, and then run SPF.
3. In the process of running SPF, or after SPF has finished running (depending on the implementation), OSPF will install new routing information in the routing table.

With the default timers, it could take up to one second (or longer, in some situations) to detect the link failure and then about three and a half seconds to flood the new information. Finally, it could take up to two and a half seconds before the receiving routers will run SPF and install the new routing information. With faster times and various sorts of tuning, you can decrease these numbers to about one second or even in the 300-millisecond range in some specific deployments.

Making Router D an area border router (ABR) dramatically impacts the convergence time from the Router E perspective because Router D has to perform all the preceding steps to start convergence. After Router D has calculated the new correct routing information, it must generate and flood a new summary LSA to Router E, and Router E has to recalculate SPF and install new routes.

Redistributing 10.1.1.0/24 into the network and making the area that contains Routers A, B, C, and D into a not-so-stubby area (NSSA) throws another set of timers into the problem. Router D now has to translate the Type 7 external LSA into an external Type 5 LSA before it can flood the new routing information to Router E.

These conditions do not even include the impact of multiple routes on the convergence process. EIGRP, for instance, can switch from its best path to a known loop-free path for 10,000 routes just about as fast as it can switch 1 route under similar conditions. OSPF performance is adversely impacted by the addition of 10,000 routes into the network, possibly doubling convergence time.

You can see, then, that it is not so simple to say, "EIGRP will always converge faster than OSPF," "IS-IS will always converge faster than EIGRP," or any other combination you can find. Some people say that OSPF always converges faster than EIGRP, for instance, but they are generally considering only intrarea convergence and not the impact of interarea operations, the impact of various timers, the complexity of the SPF tree, and other factors. Some people say that EIGRP always converges faster than any link-state protocol, but that depends on the number of routers involved in the convergence event. The shorter the query path, the faster the network converges.

If you align all the protocol convergence times based on the preceding examination, you generally find the convergence times in this order, from shortest to longest:

1 EIGRP with feasible successors.

2 Intrarea OSPF or IS-IS with fast or tuned timers.

3a EIGRP without feasible successors.

3b Intrarea OSPF or IS-IS with standard timers.

3c Interarea OSPF or IS-IS.

The last three are highly variable, in reality. In any particular network, OSPF, IS-IS, and EIGRP without feasible successors might swap positions on the list. The network design, configuration, and a multitude of other factors impact the convergence time more than the routing protocol does. You get the best convergence time out of a routing protocol if you play the network design to the strengths of the protocol.

Which Designs Play to the Strength of Each Protocol?

The natural question, after you have decided that network design plays into the suitability of the protocol (you have seen this to be the case for convergence speed, but the same is also true of any other factor you might consider for a given routing protocol, including management, troubleshooting, configuration, and so on) is this:

> What sorts of network designs play into the strengths of any given routing protocol?

This is not an easy question to answer because of the numerous ways to design a network that works. Two- and three-layer network designs, switched cores versus routed cores, switched user access versus routed user access—the design possibilities appear to be endless. To try to put a rope around this problem, the sections that follow examine only a few common topological elements to illustrate how to analyze a specific topology and design and try to determine how a routing protocol will react when running on it.

The specific types of network topologies considered here are as follows:

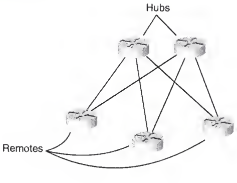
- Hub-and-spoke designs
- Full mesh designs
- Highly redundant designs

After you consider each of these specific topology elements, you learn the general concepts of hierarchical network design and how each protocol plays against them.

Hub-and-Spoke Topologies

Hub-and-spoke network designs tend to be simple in theory and much harder in implementation. Scaling tends to be the big problem for hub-and-spoke topologies. The primary focus here is the capability of a routing protocol to maintain a multitude of routing neighbors and to converge to massive network events in an acceptable amount of time. Assume, throughout this section, that you are always dealing with dual-homed hub-and-spoke networks, as Figure G-2 illustrates.

Figure G-2 *Dual-Homed Hub-and-Spoke Network*



Start by considering the following simple question:

> How many spokes or remote routers does it take to really start stressing any routing protocol that is running over a hub-and-spoke network design?

The answer to this question always depends on various factors, including link speed and stability, router processing speed and packet switching speeds, and other factors. However, general experience shows that a high-speed router (in terms of processing power) with reasonably good design supports at least 100 remote sites with any modern routing protocol.

When considering network designs in which hundreds of remote sites are available, however, you need to use special techniques with each protocol to scale the number of remote sites attached to a single pair of hub routers. Look at each protocol to see what types of problems you might encounter and what types of tools are available to resolve those problems:

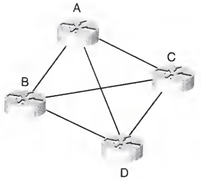
- OSPF floods topology information to each router within an area and summaries of reachability information into the area. You can place all the remote site routers into one or more OSPF *stub areas*, which cuts down on the amount of information flooded out to each remote site. Any change on a remote site is still flooded to every other remote site within the same area. For that reason, the design becomes a tradeoff between the number of areas that you want to manage and that the hub routers support and the amount of information that you can flood through the low-speed links connecting the remote stub sites.
- IS-IS also floods information to each router within an area. It does not, by default, flood information from the core of the network (the L2 routing domain) into each area. Again, you still face the tradeoff of how many level 1 routing domains you want to support at the hub routers versus how much information you can flood toward each remote router.
- The primary factor in determining scaling and convergence time in an EIGRP hub-and-spoke network is the number of queries the hub router needs to generate or process when the network changes, and the number of updates the hub router needs to generate toward the remote. Normally, if a hub loses several routes, for instance, it needs to generate queries for each of those routes to each of the remote sites. The remote sites then query the other hub router, which must process and reply to each of the queries. If the number of routes is high, this can be a processor- and memory-intensive task, causing the network to converge slowly, especially if the links between the remote sites and the hub routers are low speed. In this situation, you can summarize routers at the core toward the remote routers and block the routing information transmitted up toward the core routers. You can also cut down on the query range into the hub-and-spoke network dramatically. EIGRP, however, also provides a special operational mode for the remote sites; you can configure the remote sites as *stubs*, which indicates to the hub routers that the remote sites are never used for transiting traffic. If the remote sites are configured as stub routers, the hub router never queries them for lost routes, and the scaling properties change dramatically.

EIGRP, in theory, scales much better in a hub-and-spoke topology—and this is true in real networks, too. You often find EIGRP hub-and-spoke networks that have more than 500 remote sites attached to a pair of hub routers, over low bandwidth links, in the wild. In contrast, you tend to see OSPF and IS-IS hub-and-spoke networks top out at around 200 remote sites, even if higher bandwidth links are involved.

Full Mesh Topologies

Full mesh topologies are a less common design element in networks, but they are worth considering because the scaling properties of a routing protocol in a full mesh design indicate, to some degree, the scaling properties of the same protocol in a partial mesh design. You can think of a full mesh topology as a special case of a partial mesh topology. Again, look at the challenges and tools that are available for each protocol. Use the network illustrated in Figure G-3 throughout this discussion.

Figure G-3 *Full Mesh Network*



- Each OSPF router sends topology information to each adjacent neighbor within an area (flooding domain). If Router A receives a new link-state advertisement (LSA), Router D receives three copies of this new LSA: one from Router A, one from Router B, and one from Router C. The Cisco IOS Software implementation of OSPF does have an option to control the flooding through a full mesh network, using the **database filter-out** command.
- IS-IS is similar to OSPF; each router sends topology information to each adjacent neighbor. Cisco IOS Software enables you to control flooding through *mesh groups*.
- Each router in an EIGRP network sends each of the routes it is using to forward traffic to each neighbor. In this network, Router D is going to receive three copies of any new routing information that Router A receives, one copy from Router A, one from Router B, and one from Router C. These three copies of the routing information might be the same, but they indicate reachability through three different next hops (or neighbors). Reducing the information propagated through the mesh is difficult, at best. You can filter these routing updates through some paths within the mesh to decrease the amount of information flooded through the mesh, but that also reduces the number of paths usable through the mesh for any specific destination.

OSPF and IS-IS flood extra information through a mesh topology by default, but you can use tools to reduce the amount of flooding in highly meshed topologies. EIGRP sends updates through each router in thc mcsh, but it is difficult to reduce the number of these updates unless you want to decrease the number of paths that the network actually uses through the mesh.

In the real world, OSPF and IS-IS scale better in highly meshed environments, especially if you implement flooding reduction techniques. This is a matter of scale, of course; networks that have a mesh network of 20 or 30 routers work fine with any of the three routing protocols. However, when the mesh starts surpassing this number of routers, the special techniques that OSPF and IS-IS offcr to scale further can make a diffcrence.

Interaction with Hierarchical Designs

Traditional network design is based on layers, either two or three, that abstract the network details into "black boxes" and divide functionality vertically through the network to make management and design easier:

- The two-layer model has *aggregation* and *core layers*, or *areas*, within the network.
- The three-layer model has *access*, *distribution*, and *core layers*.

How do these layered network designs interact with each protocol? Consider each protocol in turn.

OSPF splits flooding domains into areas that are separated by ABRs. Because every router within an area must share the same link-state database to calculate loop-free paths through the network, the only place that route aggregation can be performed is at an ABR. ABRs actually aggregate two types of information:

- Information about the topology of an area that is hidden from other areas at these border edges
- Aggregation of reachability information that can be configured at these border edges

This combination of route aggregation points and flooding domain boundaries in the network implies several things:

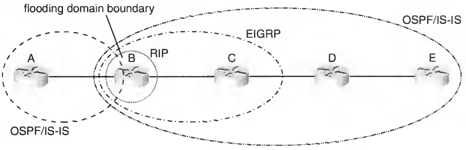
- In all three-layer network designs with OSPF, you should place the ABR in the distribution layer of the nctwork.
- In all two-layer network designs with OSPF, you should place the ABR at the aggregation to core layer edge of the network.
- Thc most aggregation points that you can cross when passing from one edge of the network to the opposite edge of the network is two.

These topological limitations might not be major in smaller networks, but in networks that have thousands of routers, they could impose severe restrictions on the network design. Network designers and operators normally break up OSPF networks at this size into multiple administrative domains, connecting the separate domains through BGP or some other mechanism.

IS-IS is similar to OSPF in its restrictions, cxcept that IS-IS allows the core and outlying flooding domains to overlap. This introduces a degree of flexibility that OSPF does not provide, but you can still only aggregate routing information at thc edges whcrc two flooding domain meet, and you cannot build more than two levels of routing into the network.

EIGRP, as a distance vector protocol, does not divide the concepts of topology summarization and routing aggregation; topology beyond one hop away is hidden by the natural operation of the protocol. Figure G-4 illustrates thc conceptual difference among EIGRP, OSPF/IS-IS, and RIP in terms of topology information propagated through the network.

Figure G-4 *Topological Awareness in Routing Protocols*



If you examine the scope through which routing information is transmitted (or known) within a nctwork, you find the following:

- The Bellman-Ford algorithm, used by the Routing Information Protocol (RIP) and the Interior Gateway Routing Protocol (IGRP), uscs only information about the local cost to reach a given destination. If Router B is running RIP, it considers only the total cost of the path to reach a destination at Router E when deciding on the bcst (loop-frcc) path.
- Diffusing Update Algorithm (DUAL), used by EIGRP, considers the local cost to reach a given destination and the cost of each neighbor to reach the same destination when calculating which available paths are loop free. EIGRP uses an awareness of the topology that is one hop away from the calculating router.
- OSPF and IS-IS, which are link-state protocols, do not use information about thc mctrics of a neighbor; rather, they count on being aware of the entire topology when calculating a loop-free path. At a flooding domain border, OSPF and IS-IS act much like distance vector protocols. Router A does not know about the topology behind Router B; it only knows the cost of Router B to reach destinations that are attached to Router E.

Because topology information is hidden in the natural processing of EIGRP routing updates, EIGRP is not restricted in where it can aggregate routing information within the network. This provides a grcat deal of flexibility to network designcrs who are running EIGRP. Multiplc layers of aggregation can be configured in the network. This means that moving from one edge of the network to the opposite edge of the network could mean encountering many more than two aggregation points.

The practical result of the EIGRP capability to aggregate routing information anywhere in the network is that many existing large-scale (2000 router and larger) networks run within a single EIGRP process or administrative domain. The feasibility of building networks this large is based on the capability to use route aggregation to divide the network into multiple layers, or sections, each acting fairly independently of the other. Although it is possible to build an OSPF or IS-IS network this large, designing and managing this network is more difficult because of thc restrictions that link-state protocols place on aggregation points.

In general, up to some relative size, the protocols are relatively equal in their capability to work with hierarchical network designs. OSPF and IS-IS tend to be less flexible about where route aggregation can be placed in the network, making it more difficult, in some situations, to fit the network design and the protocol design together. EIGRP excels at fitting into hierarchical network design.

Topological Rules of Thumb

After examining these various network topologies and how each routing protocol tends to react, you can see that when a network does not reach the edge of a specific protocol capability on any given topology, any of the routing protocols is fine. If your network has a specific predominant topology type, however, such as large-scale hub-and-spoke or large-scale full mesh topologies, choosing a protocol to fit those topologies makes sense. You can always compromise in complex areas of your network design by making effective and stable topological design areas in which the routing protocol is really stretched to the edge of its capabilities.

What Are the Tradeoffs?

In many networks, the final decision of which routing protocol is "best" comes down to these issues:

- **Convergence speed**—How important is convergence speed? How much flexibility do you have in the design of your network around convergence speeds?
- **Predominant topologies**—Does your network design have one dominant type of topology? Would a full mesh or large-scale hub-and-spoke topology benefit from running one protocol over another?
- **Scaling strategy**—Does your scaling strategy call for dividing the network into multiple pieces, or does it call for a single IGP domain, with the network broken up into pieces through route aggregation and other techniques?
- **Maintenance and management**—Which routing protocol fits the network management style of your day-to-day operations? Which one seems easier to troubleshoot and manage in your environment?

Beyond the technical factors are some nontechnical ones. For instance, if you decide to switch protocols, what is the cost for the long term? You need to consider training costs, the cost of rcvised procedures, design cffort, and possible downtime while you convert the network from one protocol to another.

In some situations, this might not be an issue. For instance, if two networks are being merged because of a corporate merger, and each runs a different protocol, the decision might be more open to consideration. If you are going to need to convert one half of the network or the other, you can more carefully consider the technical considerations and make a decision based on those considerations alone. Howcver, if your network is stable today, you should think twice about switching protocols unless a change in the business environment or some major shift in the way the network is built indicates it is an important move to makc to mect the necds of the enterprise.